CHAPTER 41

Configuring QoS

This chapter describes how to configure quality of service (QoS) on the Catalyst 6000 family switches and includes the configuration information required to support Common Open Policy Service (COPS) and Resouce ReSerVation Protocol (RSVP).

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6000 Family Command Reference* publication.

You can configure QoS using one of the following:

- SNMP
- COPS protocol
- RSVP null service template and receiver proxy functionality
- Command-line interface (CLI)

This chapter consists of these sections:

Note On the Catalyst 6000 family switches, queue architecture and QoS queuing features such as Weighted-Round Robin (WRR) and Weighted Random Early Detection (WRED) are implemented with a fixed configuration in Application Specific Integrated Circuits (ASICs); they cannot be reconfigured to different queue structures or different dequeuing methods.

Understanding How QoS Works

Note Throughout this publication and all Catalyst 6000 family documents, the term “QoS” refers to the QoS feature as implemented on the Catalyst 6000 family.

Typically, networks operate on a *best-effort* delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

The QoS feature on the Catalyst 6000 family switches selects network traffic, prioritizes it according to its relative importance, and provides priority-indexed treatment through congestion avoidance techniques. Implementing QoS in your network makes network performance more predictable and bandwidth utilization more effective.

QoS sets Layer 2 and Layer 3 values in network traffic to a configured value or to a value based on received Layer 2 or Layer 3 values. IP traffic retains the Layer 3 value when it leaves the switch.

These sections describe QoS:

Definitions

This section defines some QoS terminology:

- *Packets* carry traffic at Layer 3.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *Labels* are prioritization values carried in packets and frames:
  - Layer 2 class of service (CoS) values range between zero for low priority and seven for high priority:

    Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p CoS value in the three least significant bits.

    Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most significant bits, which are called the User Priority bits.

    Other frame types cannot carry CoS values.

Note On ports configured as ISL trunks, all traffic is in ISL frames. On ports configured as 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

  - Layer 3 IP precedence values—The IP version 4 specification defines the three most significant bits of the 1-byte Type of Service (ToS) field as IP precedence. IP precedence values range between zero for low priority and seven for high priority.
  - Layer 3 differentiated services code point (DSCP) values—The Internet Engineering Task Force (IETF) defines the six most significant bits of the 1-byte ToS field as the DSCP. The priority represented by a particular DSCP value is configurable. DSCP values range between 0 and 63 (for more information, see the "Configuring DSCP Value Maps" section on page 41-55).

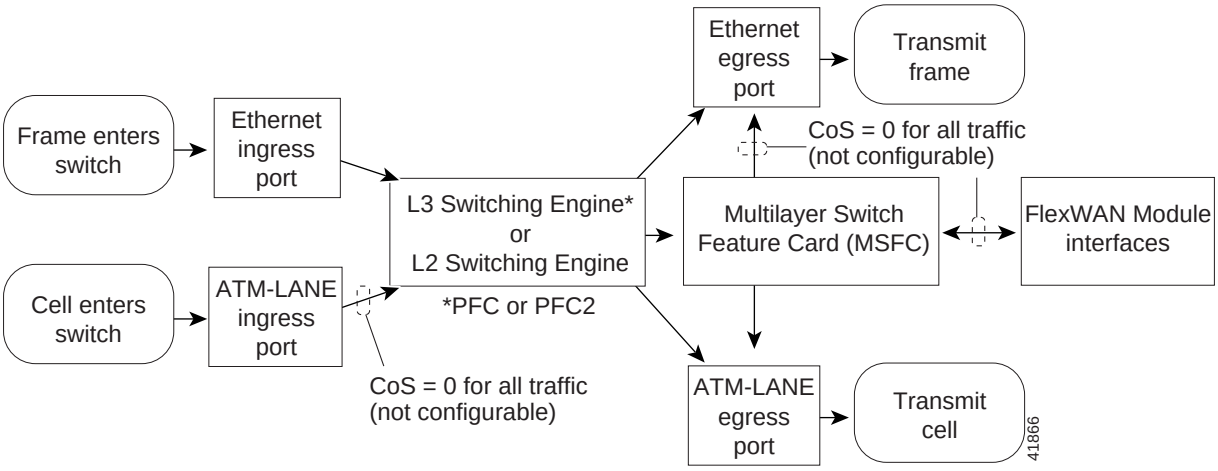
Note Layer 3 IP packets can carry either an IP precedence value or a DSCP value. QoS supports the use of either value, because DSCP values can be set equal to IP precedence values.

- *Classification* is the selection of traffic to be marked.
- *Marking*, according to RFC 2475, is the process of setting a Layer 3 DSCP value in a packet; in this publication, the definition of marking is extended to include setting Layer 2 CoS values.
- *Scheduling* is the assignment of traffic to a queue. QoS assigns traffic based on CoS values.
- *Congestion avoidance* is the process by which QoS reserves ingress and egress port capacity for traffic with high-priority CoS values. QoS implements congestion avoidance with CoS value-based drop thresholds. A drop threshold is the percentage of buffer utilization at which traffic with a specified CoS value is dropped, leaving the buffer available for traffic with higher-priority CoS values.
- *Policing* is the process by which the switch limits the bandwidth consumed by a flow of traffic. Policing can mark or drop traffic.
- Except where specifically differentiated, *Layer 3 switching engine* refers to either:
  - Supervisor Engine 2 with Layer 3 Switching Engine II (Policy Feature Card 2 or PFC2)
  - Supervisor Engine 1 with Layer 3 Switching Engine WS-F6K-PFC (Policy Feature Card or PFC)

Flowcharts

Figure 41-1 shows how traffic flows through the QoS features; Figure 41-2 through Figure 41-7 show more details of the traffic flow through QoS features.

Figure 41-1 Traffic Flow Through QoS Features



Note Traffic that is Layer 3 switched does not go through the Multilayer Switch Feature Card (MSFC or MSFC2) and retains the CoS value assigned by the Layer 3 switching engine.

Note Enter the **show port capabilities** command to see the queue structure of a port (for more information, see the "Receive Queues" section on page 41-11 and the "Transmit Queues" section on page 41-25).

Figure 41-2 Ethernet ingress Port Classification, Marking, Scheduling, and Congestion Avoidance

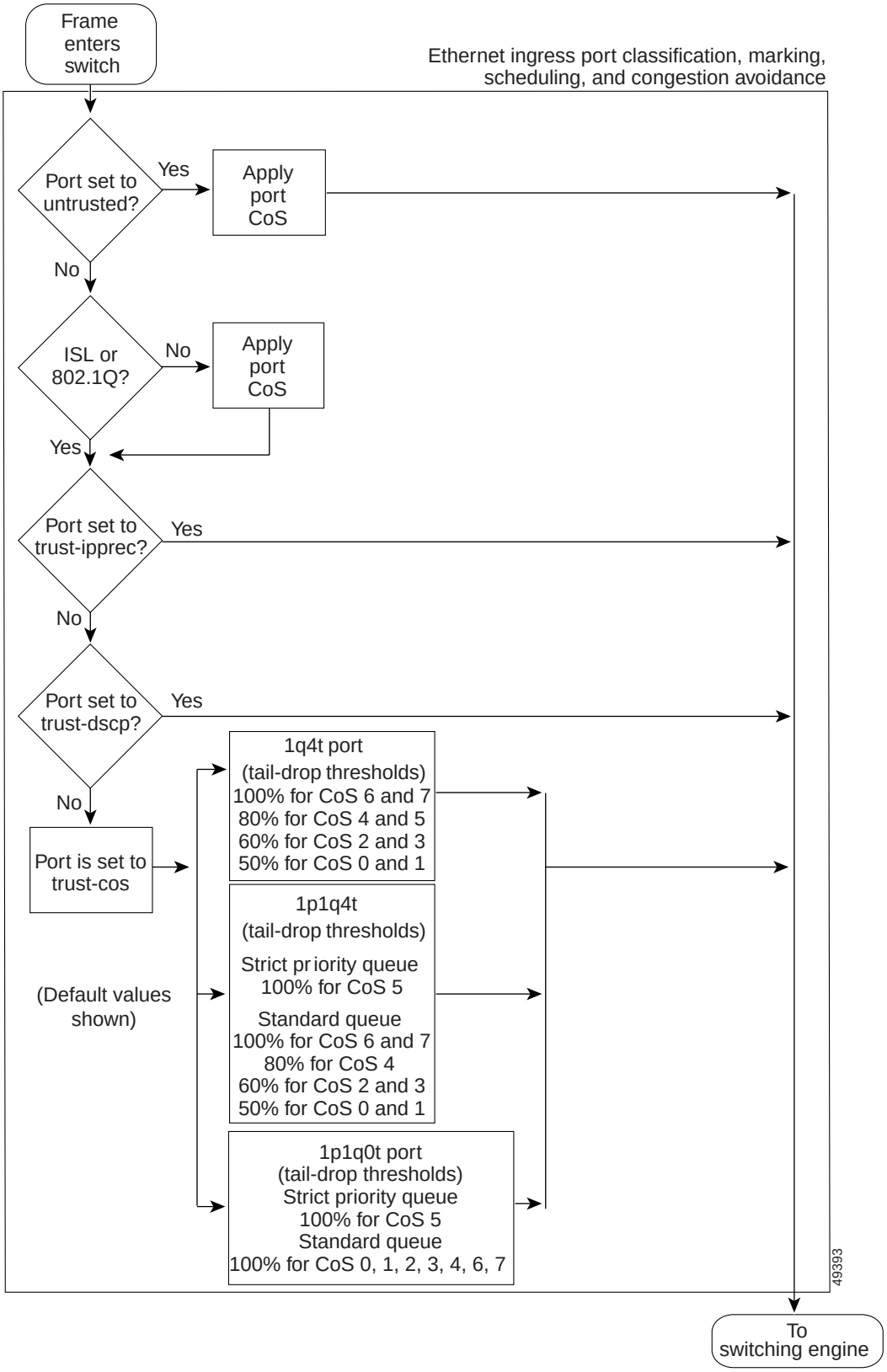
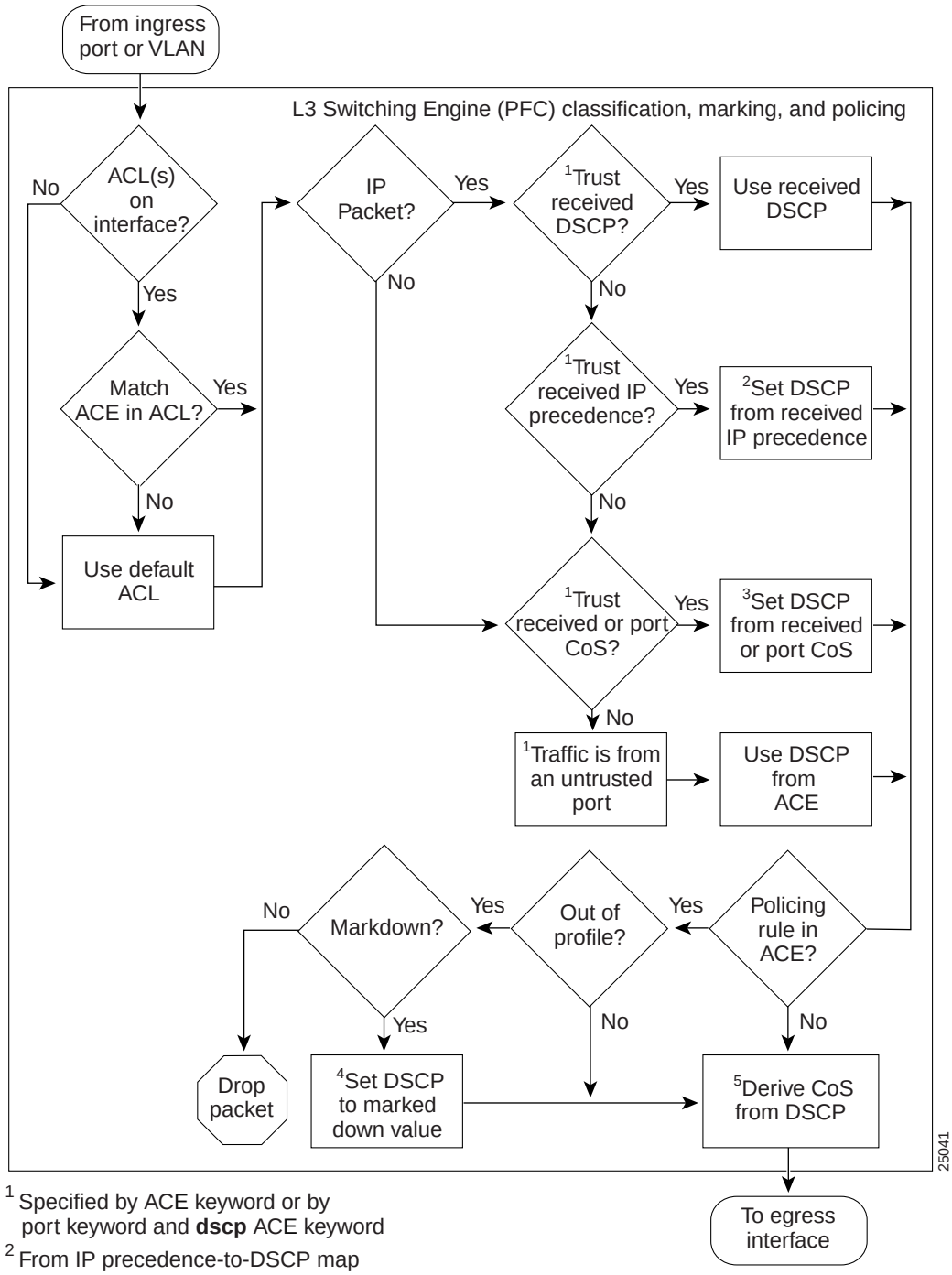


Figure 41-3 Layer 3 Switching Engine Classification, Marking, and Policing



[1] Specified by ACE keyword or by port keyword and **dscp** ACE keyword

[2] From IP precedence-to-DSCP map

[3] From CoS-to-DSCP map

[4] From DSCP markdown map

[5] From DSCP-to-CoS map

Figure 41-4 Layer 2 Switching Engine Classification and Marking

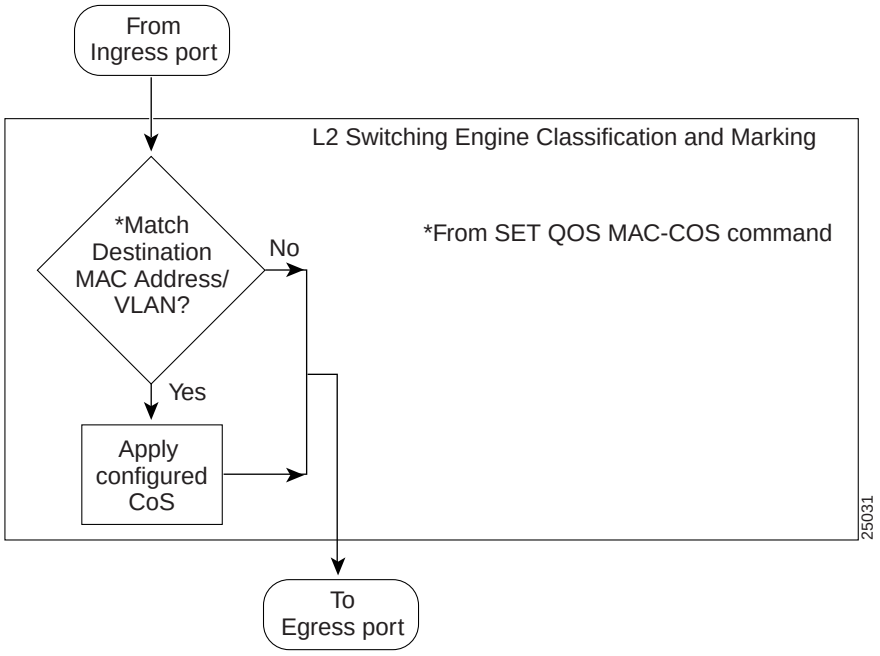


Figure 41-5 Multilayer Switch Feature Card Marking (MSFC and MSFC2)

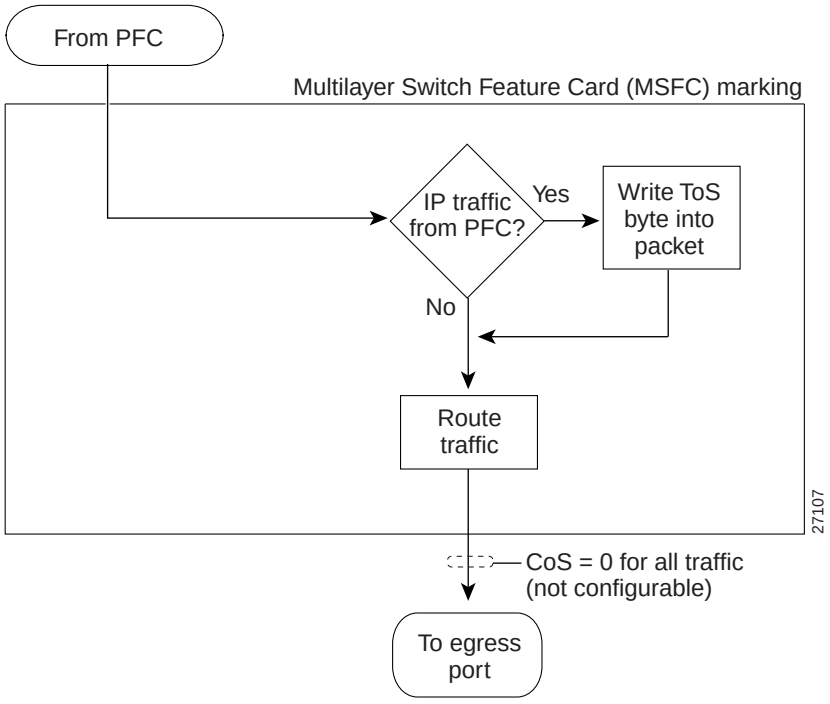


Figure 41-6 Ethernet Egress Port Scheduling, Congestion Avoidance, and Marking

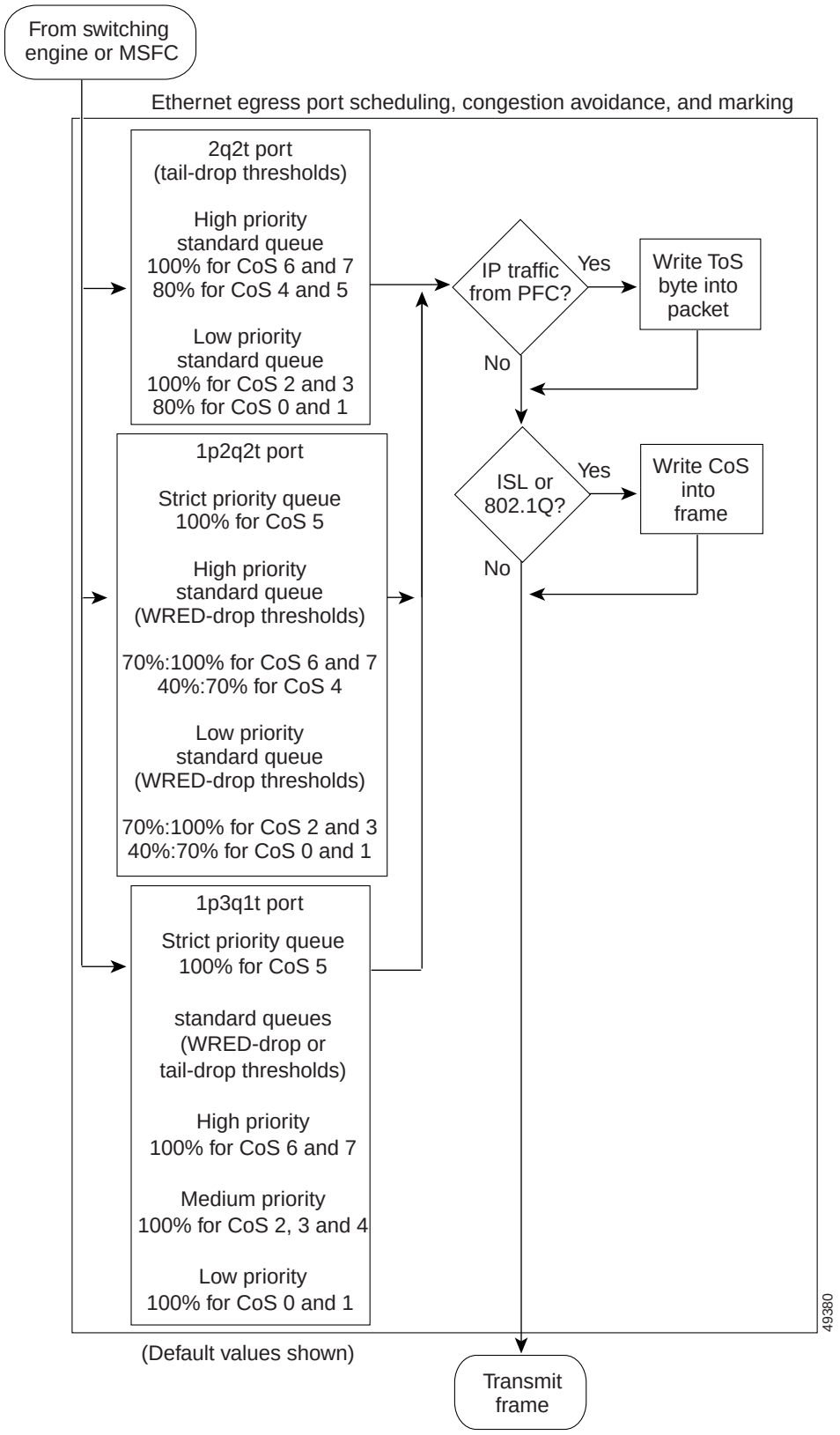
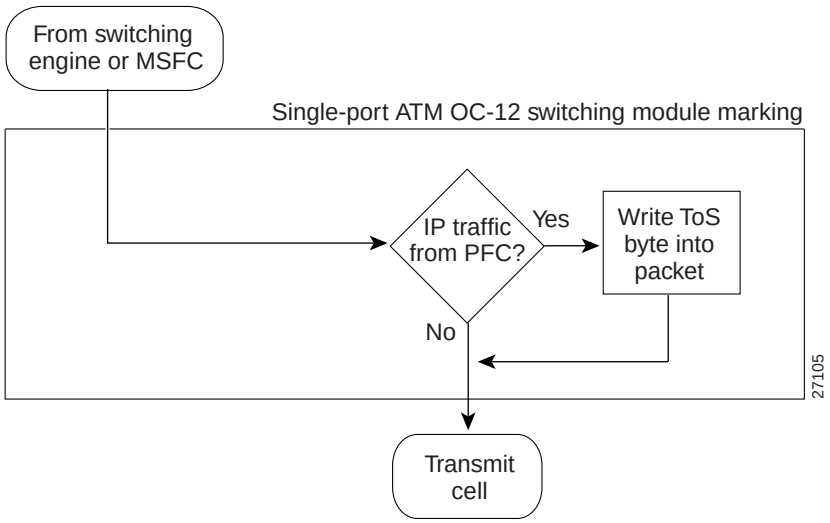


Figure 41-7 Single-Port ATM OC-12 Switching Module Marking



QoS Feature Set Summary

The QoS feature set on your switch is determined by the switching engine on the supervisor engine. Enter the **show module** command for the supervisor engine to display your switching engine configuration. The display shows the "Sub-Type" to be one of the following:

- Supervisor Engine 2 (WS-X6K-SUP2-2GE) with Layer 3 Switching Engine II (WS-F6K-PFC2—Policy Feature Card 2 or PFC2)
- Supervisor Engine 1 (WS-X6K-SUP1A-2GE or WS-X6K-SUP1-2GE) with one of the following:
  - Layer 3 Switching Engine (WS-F6K-PFC—Policy Feature Card or PFC)
  - Layer 2 Switching Engine II (WS-F6020A)
  - Layer 2 Switching Engine I (WS-F6020)

The Layer 3 Switching Engine WS-F6K-PFC and Layer 3 Switching Engine II support similar feature sets. The two Layer 2 switching engines support the same QoS feature set.

These sections describe the QoS feature sets:

- Ethernet Ingress Port Features, page 41-9
- Layer 3 Switching Engine Features, page 41-9
- Layer 2 Switching Engine Features, page 41-9
- Ethernet Egress Port Features, page 41-9
- Single-Port ATM OC-12 Switching Module Features, page 41-9
- Multilayer Switch Feature Card (MSFC or MSFC2), page 41-9

Ethernet Ingress Port Features

With any switching engine, QoS supports classification, marking, scheduling, and congestion avoidance using Layer 2 CoS values at Ethernet ingress ports. Classification, marking, scheduling, and congestion avoidance at Ethernet ingress ports do not use or set Layer 3 IP precedence or DSCP values. With a Layer 3 switching engine, you can configure Ethernet ingress port trust states that can be used by the switching engine to set Layer 3 IP precedence or DSCP values and the Layer 2 CoS value. For more information, see the "Ethernet Ingress Port Marking, Scheduling, Congestion Avoidance, and Classification" section on page 41-10.

Layer 3 Switching Engine Features

With a Layer 3 switching engine, QoS supports classification, marking, and policing using IP, IPX, and Media Access Control (MAC) access control lists (ACLs). ACLs contain access control entries (ACEs) that specify Layer 2, 3, and 4 classification criteria, a marking rule, and policing rules. Marking sets the Layer 3 IP precedence or DSCP values and the Layer 2 CoS value to either received or configured Layer 2 or Layer 3 values. Policing uses bandwidth limits to either drop or mark nonconforming traffic. For more information, see the "Classification, Marking, and Policing with a Layer 3 Switching Engine" section on page 41-14.

During processing, a Layer 3 switching engine associates a DSCP value with all traffic, including non-IP traffic (for more information, see the "Internal DSCP Values" section on page 41-15).

Layer 2 Switching Engine Features

With a Layer 2 Switching Engine, QoS can classify traffic using Layer 2 destination MAC addresses, VLANs, and marking using Layer 2 CoS values. Classification and marking with a Layer 2 Switching Engine do not use or set Layer 3 IP precedence or DSCP values. For more information, see the "Classification and Marking with a Layer 2 Switching Engine" section on page 41-24.

Ethernet Egress Port Features

With any switching engine, QoS supports Ethernet egress port scheduling and congestion avoidance using Layer 2 CoS values. Ethernet egress port marking sets Layer 2 CoS values and, with a Layer 3 switching engine, Layer 3 DSCP values. For more information, see the "Ethernet Egress Port Scheduling, Congestion Avoidance, and Marking" section on page 41-24.

Single-Port ATM OC-12 Switching Module Features

The ingress interface from a single-port ATM OC-12 switching module is untrusted, and QoS sets CoS to zero in all traffic received from it. With a Layer 3 switching engine, QoS can mark IP traffic transmitted to a single-port ATM OC-12 switching module with Layer 3 DSCP values.

Multilayer Switch Feature Card (MSFC or MSFC2)

QoS marks IP traffic transmitted to an MSFC with Layer 3 DSCP values. CoS is zero in all traffic sent from an MSFC to egress ports.

Note Traffic that is Layer 3 switched does not go through the MFSC and retains the CoS value assigned by the Layer 3 switching engine.

Ethernet Ingress Port Marking, Scheduling, Congestion Avoidance, and Classification

These sections describe Ethernet ingress port marking, scheduling, congestion avoidance, and classification:

Overview

The trust state of an Ethernet port determines how it marks, schedules, and classifies received traffic, and whether or not congestion avoidance is implemented. You can configure the trust state of each port with one of these keywords:

- **untrusted** (default)
- **trust-ipprec** (Layer 3 switching engine only—not supported on **1q4t** ports except Gigabit Ethernet)
- **trust-dscp** (Layer 3 switching engine only—not supported on **1q4t** ports except Gigabit Ethernet)
- **trust-cos**

Note On **1q4t** ports (except Gigabit Ethernet), the **trust-cos** port keyword displays an error message, activates receive queue drop thresholds, and—as indicated by the error message—does not apply the **trust-cos** trust state to traffic. You must configure the **trust-cos** ACL that matches the ingress traffic to apply the **trust-cos** trust state.

For more information, see the "Configuring the Trust State of a Port" section on page 41-32.

In addition to the port configuration keywords listed above, with a Layer 3 switching engine, QoS uses **trust-ipprec**, **trust-dscp**, and **trust-cos** ACE keywords. Do not confuse the ACE keywords with the port keywords.

Ports configured with the **untrusted** keyword are called untrusted ports. Ports configured with the **trust-ipprec**, **trust-dscp**, or **trust-cos** keywords are called trusted ports. QoS implements ingress port congestion avoidance only on ports configured with the **trust-cos** keyword.

Ingress port marking, scheduling, and congestion avoidance use Layer 2 CoS values. Ingress port marking, scheduling, and congestion avoidance do not use or set Layer 3 IP precedence or DSCP values.

Marking at Untrusted Ports

QoS marks all frames received through untrusted ports with the port CoS value (the default is zero). QoS does not implement ingress port congestion avoidance on untrusted ports: the traffic goes directly to the switching engine.

Marking at Trusted Ports

When an ISL frame enters the switch through a trusted port, QoS accepts the three least significant bits in the User field as a CoS value. When an 802.1Q frame enters the switch through a trusted port, QoS accepts the User Priority bits as a CoS value. QoS marks all traffic received in other frame types with the port CoS value.

The port CoS value is configurable for each Ethernet port (for more information, see the "Configuring the CoS Value for a Port" section on page 41-33).

Ethernet Ingress Port Scheduling and Congestion Avoidance

QoS does not implement ingress port congestion avoidance on ports configured with the **untrusted**, **trust-ipprec**, or **trust-dscp** keywords: the traffic goes directly to the switching engine.

QoS uses CoS-value-based receive-queue drop thresholds to avoid congestion in traffic entering the switch through a port configured with the **trust-cos** keyword (for more information, see the "Configuring the Trust State of a Port" section on page 41-32).

Receive Queues

Enter a **show port capabilities** command to see the queue structure of a port. The command displays one of the following:

- **rx-(1p1q4t)**—one strict-priority queue and one standard queue with four thresholds
- **rx-(1q4t)**—one standard queue with four thresholds
- **rx-(1p1q0t)**—one strict-priority queue and one standard queue with no configurable thresholds

Strict-priority queues are serviced in preference to other queues. QoS services traffic in a strict-priority queue before servicing the standard queue. When QoS services the standard queue, after receiving a packet, it checks for traffic in the strict-priority queue. If QoS detects traffic in the strict-priority queue, it suspends its service of the standard queue and completes service of all traffic in the strict-priority queue before returning to the standard queue.

Ingress Scheduling

QoS schedules traffic through the receive queues based on CoS values. In the **1p1q4t** and **1p1q0t** default configurations, QoS assigns all traffic with CoS 5 to the strict-priority queue; QoS assigns all other traffic to the standard queue. In the **1q4t** default configuration, QoS assigns all traffic to the standard queue.

Ingress Congestion Avoidance

If a port is configured with the **trust-cos** keyword, QoS implements CoS-value-based receive-drop thresholds to avoid congestion in received traffic.

1q4t ports have this default drop-threshold configuration:

- Using receive-queue drop threshold 1, the switch drops incoming frames with CoS 0 or 1 when the receive-queue buffer is 50 percent or more full.
- Using receive-queue drop threshold 2, the switch drops incoming frames with CoS 2 or 3 when the receive-queue buffer is 60 percent or more full.

- Using receive-queue drop threshold 3, the switch drops incoming frames with CoS 4 or 5 when the receive-queue buffer is 80 percent or more full.
- Using receive-queue drop threshold 4, the switch drops incoming frames with CoS 6 or 7 when the receive-queue buffer is 100 percent full.

1p1q4t ports have this default drop-threshold configuration:

- Frames with CoS 0, 1, 2, 3, 4, 6, or 7 go to the standard receive queue:
  - Using standard receive-queue drop threshold 1, the switch drops incoming frames with CoS 0 or 1 when the receive-queue buffer is 50 percent or more full.
  - Using standard receive-queue drop threshold 2, the switch drops incoming frames with CoS 2 or 3 when the receive-queue buffer is 60 percent or more full.
  - Using standard receive-queue drop threshold 3, the switch drops incoming frames with CoS 4 when the receive-queue buffer is 80 percent or more full.
  - Using standard receive-queue drop threshold 4, the switch drops incoming frames with CoS 6 or 7 when the receive-queue buffer is 100 percent full.
- Frames with CoS 5 go to the strict-priority receive queue (queue 2), where the switch drops incoming frames only when the strict-priority receive-queue buffer is 100 percent full.

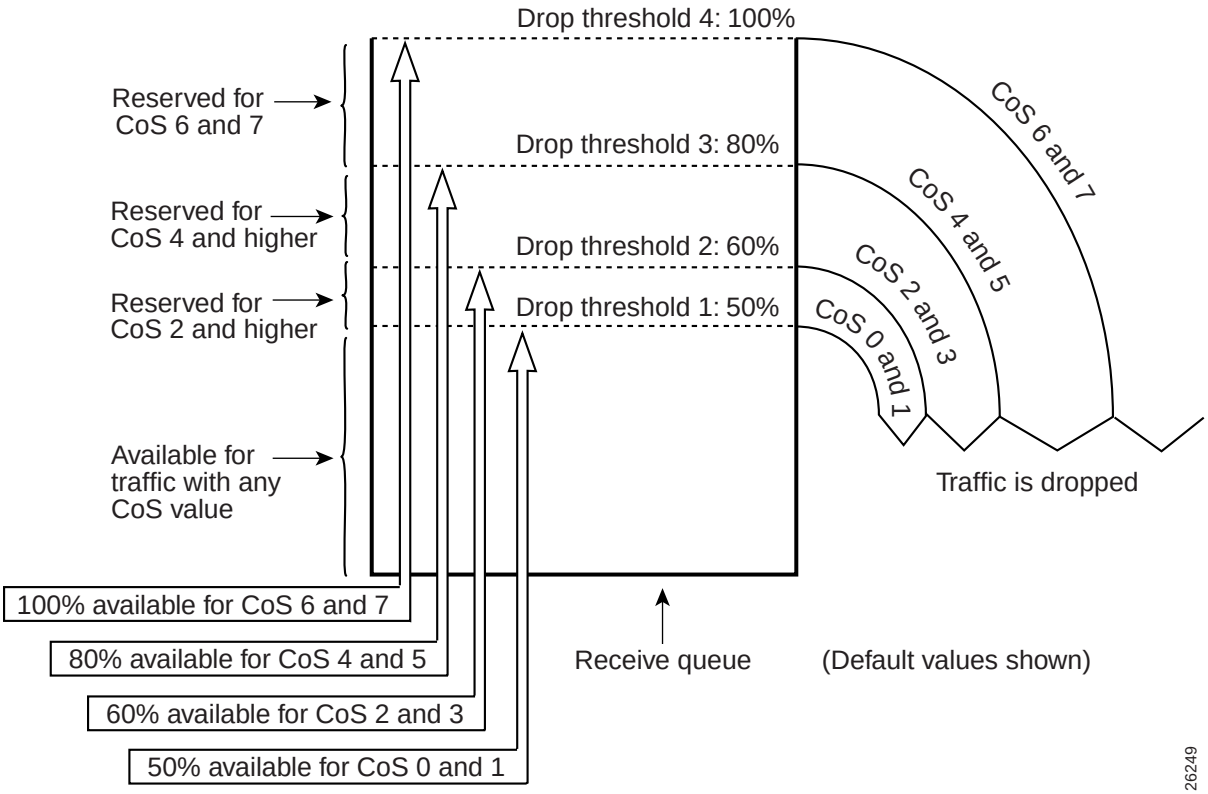
1p1q0t ports have this default drop-threshold configuration:

- Frames with CoS 0, 1, 2, 3, 4, 6, or 7 go to the standard receive queue. The switch drops incoming frames when the receive-queue buffer is 100 percent full.
- Frames with CoS 5 go to the strict-priority receive queue (queue 2), where the switch drops incoming frames only when the strict-priority receive-queue buffer is 100 percent full.

Note The explanations in this section use default values. You can configure many of the parameters (for more information, see the "Configuring QoS" section on page 41-30). All ports of the same type use the same drop-threshold configuration.

Figure 41-8 shows the drop thresholds for a **1q4t** port. Drop thresholds in other configurations function similarly.

Figure 41-8 Receive Queue Drop Thresholds



Ethernet Ingress Port Classification Features with a Layer 3 Switching Engine

You can use the **untrusted**, **trust-ipprec**, **trust-dscp**, and **trust-cos** port keywords to classify traffic on a per-port basis for a Layer 3 switching engine to mark.

The **trust-ipprec** and **trust-dscp** keywords are supported only with a Layer 3 switching engine and are not supported on **1q4t** ports except Gigabit Ethernet. On **1q4t** ports (except Gigabit Ethernet), the **trust-cos** port keyword displays an error message, activates receive-queue drop thresholds, and—as indicated by the error message—does not apply the **trust-cos** trust state to traffic. You must configure the **trust-cos** ACL that matches the ingress traffic to apply the **trust-cos** trust state.

In addition to per-port classification, you can create ACEs that classify traffic on a per-packet basis (for IP and IPX traffic, see the "Named IP ACLs" section on page 41-38 and the "Creating or Modifying Named IPX ACLs" section on page 41-42) or on a per-frame basis (for other traffic, see the "Creating or Modifying Named MAC ACLs" section on page 41-43), regardless of the port configuration (see the "Marking Rules" section on page 41-21).

To mark traffic in response to per-port classification, the traffic must match an ACE that contains the **dscp** ACE keyword (see the "Marking Rules" section on page 41-21). In their default configuration, the ACEs in the default ACLs contain the **dscp** ACE keyword. Table 41-1 lists the per-port classifications and the marking rules that they invoke.

Table 41-1 Marking Based on Per-Port Classification

Port Keyword	ACE Keyword	Marking Rule
untrusted	**dscp**	Set internal and egress DSCP as specified in the ACE.
trust-ipprec	**dscp**	For IP traffic, set internal and egress DSCP from the received Layer 3 IP precedence value. For other traffic, set internal and egress from the received or port Layer 2 CoS value. **Note**—With the **trust-ipprec** port keyword, QoS uses only the IP precedence bits. If traffic with a DSCP value enters the switch through a port configured with the **trust-ipprec** port keyword, the three most significant bits of the DSCP value are interpreted as an IP precedence value; QoS ignores the rest of the DSCP value.
trust-dscp	**dscp**	For IP traffic, set internal and egress DSCP from the received Layer 3 DSCP value. For other traffic, set internal and egress DSCP from the received or port Layer 2 CoS value.
trust-cos	**dscp**	Set internal and egress DSCP from the received or port Layer 2 CoS value.

QoS uses configurable mapping tables to set internal and egress DSCP, which is a 6-bit value, from CoS and IP precedence, which are 3-bit values (for more information, see the "Internal DSCP Values" section on page 41-15 and the "Configuring DSCP Value Maps" section on page 41-55).

Classification, Marking, and Policing with a Layer 3 Switching Engine

Note With a Layer 3 switching engine, the Catalyst 6000 family switches provide QoS only for the following frame types: Ethernet_II, Ethernet_802.3, Ethernet_802.2, and Ethernet_SNAP.

These sections describe classification, marking, and policing with a Layer 3 switching engine:

Note Classification with a Layer 3 switching engine uses Layer 2, 3, and 4 values. Marking with a Layer 3 switching engine uses Layer 2 CoS values and Layer 3 IP precedence or DSCP values.

Internal DSCP Values

These sections describe the internal DSCP values:

Internal DSCP Sources

During processing, the priority of all traffic (including non-IP traffic) is represented with an internal DSCP value. QoS derives the internal DSCP value from the following:

- For **trust-cos** traffic, from received or port Layer 2 CoS values (traffic from an untrusted port has the port CoS value and if traffic from an untrusted port matches a **trust-cos** ACL, QoS derives the internal DSCP value from the port CoS value)
- For **trust-ipprec** traffic, from received IP precedence values
- For **trust-dscp** traffic, from received DSCP values
- For **untrusted** traffic, from port CoS or configured DSCP values

The trust state of traffic is the trust state of the ingress port unless set otherwise by the matching ACE.

Note A **trust-cos** ACL cannot restore received CoS in traffic from untrusted ports. Traffic from untrusted ports always has the port CoS value.

QoS uses configurable mapping tables to derive the internal 6-bit DSCP value from CoS or IP precedence, which are 3-bit values (see the"Mapping Received CoS Values to Internal DSCP Values" section on page 41-55 or the "Mapping Received IP Precedence Values to Internal DSCP Values" section on page 41-56).

Egress DSCP and CoS Sources

For egress IP traffic, QoS creates a ToS byte from the internal DSCP value (which you can set equal to an IP precedence value) and sends it to the egress port to be written into IP packets. For **trust-dscp** and **untrusted** IP traffic, the ToS byte includes the original 2 least-significant bits from the received ToS byte.

For all egress traffic, QoS uses a configurable mapping table to derive a CoS value from the internal DSCP value associated with traffic (see the "Mapping Internal DSCP Values to Egress CoS Values" section on page 41-56). QoS sends the CoS value to Ethernet egress ports for use in scheduling and to be written into ISL and 802.1Q frames.

ACLs

QoS uses ACLs that contain ACEs. The ACEs specify classification criteria, a marking rule, and policing rules. QoS compares received traffic to the ACEs in ACLs until a match occurs. When the traffic matches the classification criteria in an ACE, QoS marks and polices the packet as specified in the ACE and makes no further comparisons.

There are three ACL types: IP and, with a Layer 3 switching engine, IPX and MAC. QoS compares traffic of each type (IP, IPX, and MAC) only to the corresponding ACL type (see Table 41-2).

Table 41-2 Supported Ethertype Field Values

ACL Type	Ethertype Field Value	Protocol
IP	0x0800	IP
IPX	0x8137 and 0x8138	IPX
MAC[1]	0x0600 and 0x0601	XNS
	0x0BAD and 0x0BAF	Banyan VINES
	0x6000-0x6009 and 0x8038-0x8042	DECnet
	0x809b and 0x80f3	AppleTalk

1. QoS MAC ACLs that do not include an ethertype parameter match traffic with any value in the ethertype field, which allows MAC-level QoS to be applied to any traffic except IP and IPX.

QoS supports user-created *named* ACLs, each containing an ordered list of ACEs, and user-configurable *default* ACLs, each containing a single ACE.

Named ACLs

You create a named ACL when you enter an ACE with a new ACL name. You add an ACE to an existing ACL when you enter an ACE with the name of the existing ACL.

You can specify the classification criteria for each ACE in a named ACL. The classification criteria can be specific values or wildcards (for more information, see the "Creating or Modifying ACLs" section on page 41-37).

These sections describe the classification criteria that can be specified in a named ACL:

IP ACE Layer 3 Classification Criteria

You can create IP ACEs that match traffic with specific Layer 3 values by including these Layer 3 parameters (see the "Named IP ACLs" section on page 41-38):

- IP source address and mask, entered as specific values or with the **any** keyword or with the **host** keyword and a host address.
- IP destination address and mask, entered as specific values or with the **any** keyword or with the **host** keyword and a host address.
- DSCP value (0–63) or IP precedence specified with a numeric value (0–7) or these keywords:
  - **Network** (IP precedence 7)
  - **Internet** (IP precedence 6)

- **Critical** (IP precedence 5)
- **Flash-override** (IP precedence 4)
- **Flash** (IP precedence 3)
- **Immediate** (IP precedence 2)
- **Priority** (IP precedence 1)
- **Routine** (IP precedence 0)

Note IP ACEs that do not include a DSCP or IP precedence value parameter match all DSCP or IP precedence values.

IP ACE Layer 4 Protocol Classification Criteria

You can create IP ACEs that match specific Layer 4 protocol traffic by including a Layer 4 protocol parameter (see the "IP ACLs for Other Layer 4 Protocols" section on page 41-41). You can specify the protocol numerically (0–255) or with these keywords: **ahp** (51), **eigrp** (88), **esp** (50), **gre** (47), **igrp** (9), **icmp** (1), **igmp** (2), **igrp** (9), **ip** (0), **ipinip** (4), **nos** (94), **ospf** (89), **pcp** (108), **pim** (103), **tcp** (6), or **udp** (17).

Note IP ACEs that do not include a Layer 4 protocol parameter or that include the **ip** keyword match all IP traffic.

IP ACE Layer 4 TCP Classification Criteria

You can create Transmission Control Protocol (TCP) ACEs that match traffic for specific TCP ports by including TCP source and/or destination port parameters (for more information, see the "IP ACEs for TCP Traffic" section on page 41-39). You can specify TCP port parameters numerically (0–65535) or with these keywords:

Keyword	Port		Keyword	Port		Keyword	Port		Keyword	Port
bgp	179		**ftp**	21		**lpd**	515		**telnet**	23
chargen	19		**ftp-data**	20		**nntp**	119		**time**	37
daytime	13		**gopher**	70		**pop2**	109		**uucp**	540
discard	9		**hostname**	101		**pop3**	110		**whois**	43
domain	53		**irc**	194		**smtp**	25		**www**	80
echo	7		**klogin**	543		**sunrpc**	111			
finger	79		**kshell**	544		**tacacs**	49			

Note TCP ACEs that do not include a Layer 4 TCP port parameter match all TCP traffic.

IP ACE Layer 4 UDP Classification Criteria

You can create User Datagram Protocol (UDP) ACEs that match traffic for specific UDP source and/or destination ports by including UDP port parameters (for more information, see the "IP ACEs for UDP Traffic" section on page 41-39). You can specify UDP port parameters numerically (0–65535) or with these keywords:

Keyword	Port		Keyword	Port		Keyword	Port		Keyword	Port
biff	512		**echo**	7		**rip**	520		**talk**	517
bootpc	68		**mobile-ip**	434		**snmp**	161		**tftp**	69
bootps	67		**nameserver**	42		**snmptrap**	162		**time**	37
discard	9		**netbios-dgm**	138		**sunrpc**	111		**who**	513
dns	53		**netbios-ns**	137		**syslog**	514		**xdmcp**	177
dnsix	195		**ntp**	123		**tacacs**	49			

Note UDP ACEs that do not include a Layer 4 UDP port parameter match all UDP traffic.

IP ACE Layer 4 ICMP Classification Criteria

You can create Internet Control Management Protocol (ICMP) ACEs that match traffic containing specific ICMP messages by including ICMP types and, optionally, ICMP codes (for more information, see the "IP ACEs for ICMP Traffic" section on page 41-40). You can specify ICMP types and codes numerically (0–255) or with these keywords:

Keyword	Type	Code	Keyword	Type	Code
administratively-prohibited	3	13	**net-tos-unreachable**	3	11
alternate-address[1]	6	—	**net-unreachable**	3	0
conversion-error	31	0	**network-unknown**	3	6
dod-host-prohibited	3	10	**no-room-for-option**	12	2
dod-net-prohibited	3	9	**option-missing**	12	1
echo	8	0	**packet-too-big**	3	4
echo-reply	0	0	**parameter-problem**	12	0
general-parameter-problem[1]	12	—	**port-unreachable**	3	3
host-isolated	3	8	**precedence-unreachable**	3	15
host-precedence-unreachable	3	14	**protocol-unreachable**	3	2
host-redirect	5	1	**reassembly-timeout**	11	1
host-tos-redirect	5	3	**redirect**[1]	5	—
host-tos-unreachable	3	12	**router-advertisement**	9	0
host-unknown	3	7	**router-solicitation**	10	0
host-unreachable	3	1	**source-quench**	4	0
information-reply	16	0	**source-route-failed**	3	5

Keyword	Type	Code	Keyword	Type	Code
information-request	15	0	**time-exceeded**[1]	11	—
mask-reply	18	0	**timestamp-reply**	14	0
mask-request	17	0	**timestamp-request**	13	0
mobile-redirect	32	0	**traceroute**	30	0
net-redirect	5	0	**ttl-exceeded**	11	0
net-tos-redirect	5	2	**unreachable**[1]	3	—

1. Matches all code values

Note ICMP ACEs with only a Layer 4 ICMP *type* parameter match all *code* values for that *type* value. ICMP ACEs that do not include any Layer 4 ICMP type and code parameters match all ICMP traffic.

IP ACE Layer 4 IGMP Classification Criteria

You can create IGMP ACEs that match traffic containing specific IGMP messages by including an IGMP type parameter (for more information, see the "IP ACEs for IGMP Traffic" section on page 41-40). You can specify the IGMP type numerically (0–255) or with these keywords: **host-query** (1), **host-report** (2), **dvmrp** (3), **pim** (4), or **trace** (5).

Note QoS does not support Internet Group Management Protocol (IGMP) traffic when IGMP snooping is enabled. QoS supports IGMP classification using version 1 four-bit Type fields.

Note IGMP ACEs that do not include a Layer 4 IGMP type parameter match all IGMP traffic.

IPX ACE Classification Criteria

You can create IPX ACEs that match specific IPX traffic by including these parameters (for more information, see the "Creating or Modifying Named IPX ACLs" section on page 41-42):

- IPX source network (-1 matches any network number)
- Protocol, which can be specified numerically (0–255) or with these keywords: **any**, **ncp** (17), **netbios** (20), **rip** (1), **sap** (4), **spx** (5)
- IPX ACEs support the following optional parameters:
  - IPX destination network (-1 matches any network number)
  - If you specify an IPX destination network, IPX ACEs support the following optional parameters: an IPX destination network mask (-1 matches any network number), an IPX destination node, and an IPX destination node mask

MAC ACE Layer 2 Classification Criteria

You can create MAC ACEs that match specific Ethernet traffic by including these Layer 2 parameters (for more information, see the "Creating or Modifying Named MAC ACLs" section on page 41-43):

- Ethernet source and destination addresses and masks, entered as specific values or with the **any** keyword or with the **host** keyword and a host Ethernet address
- Optionally, an ethertype parameter from this list:
  - 0x809B (or **ethertalk**)
  - 0x80F3 (or **aarp**)
  - 0x6001 (or **dec-mop-dump**)
  - 0x6002 (or **dec-mop-remote-console**)
  - 0x6003 (or **dec-phase-iv**)
  - 0x6004 (or **dec-lat**)
  - 0x6005 (or **dec-diagnostic-protocol**)
  - 0x6007 (or **dec-lavc-sca**)
  - 0x6008 (or **dec-amber**)
  - 0x6009 (or **dec-mumps**)
  - 0x8038 (or **dec-lanbridge**)
  - 0x8039 (or **dec-dsm**)
  - 0x8040 (or **dec-netbios**)
  - 0x8041 (or **dec-msdos**)
  - 0x8042 (no keyword)
  - 0x0BAD (no keyword)
  - 0x0baf (or **banyan-vines-echo**)
  - 0x0600 (or **xerox-ns-idp**)

QoS MAC ACLs that do not include an ethertype parameter match traffic with any value in the ethertype field, which allows MAC-level QoS to be applied to any traffic except IP and IPX.

Default ACLs

There are three default ACLs, one each for IP and, with a Layer 3 switching engine, IPX and MAC traffic. Each ACL has a single ACE that has a configurable marking rule and configurable policing rules. The default ACLs have nonconfigurable classification criteria that matches all traffic. QoS compares any traffic with a supported ethertype field value that does not match a named ACL to the default ACLs. Unmatched IP traffic matches the default IP ACL. Unmatched IPX traffic matches the default IPX ACL. Unmatched Ethernet traffic matches the default MAC ACL.

Note All traffic matches an ACE in an ACL, either an ACE in a named ACL or one of the default ACLs, because the default ACLs match all traffic.

Marking Rules

Note Marking is not supported for IPX or MAC traffic with a PFC2.

Marking rules specify how QoS marks traffic when the traffic matches the filtering parameters in an ACE (see the "ACE Name, Marking Rule, Policing, and Filtering Syntax" section on page 41-37). QoS supports four marking rules, specified with the following four ACE keywords: **trust-dscp**, **trust-ipprec**, **trust-cos**, and **dscp**. Each ACE contains one of the keywords. The marking rules are as follows:

- **trust-dscp** (IP ACLs only)—Instructs QoS to set internal and egress DSCP from received DSCP values (see the "Internal DSCP Values" section on page 41-15).
- **trust-ipprec** (IP ACLs only)—Instructs QoS to set internal and egress DSCP from received IP precedence values.

  Note With the **trust-ipprec** port keyword, QoS uses only the IP precedence bits. If traffic with a DSCP value enters the switch through a port configured with the **trust-ipprec** port keyword, the three most significant bits of the DSCP value are interpreted as an IP precedence value; QoS ignores the rest of the DSCP value.

- **trust-cos** (all ACLs except IPX and MAC with a PFC2)—Instructs QoS to set internal and egress DSCP from received or port CoS values. In traffic from ports configured with the **trust-cos** keyword, the CoS value is that received in ISL and 802.1Q frames; in all other cases, the CoS value is that configured on the port (default is zero).
- **dscp** (all ACLs except IPX and MAC with a PFC2)—Instructs QoS to mark traffic as indicated by the port trust keywords:
  - In IP traffic from ingress ports configured with the **trust-dscp** port keyword, the **dscp** ACE keyword instructs QoS to set the internal and egress DSCP values from the received DSCP values. In non-IP traffic, QoS sets the DSCP from the received or port CoS value.
  - In IP traffic from ingress ports configured with the **trust-ipprec** port keyword, the **dscp** ACE keyword instructs QoS to set the internal and egress DSCP values from the received IP precedence values. In non-IP traffic, QoS sets the DSCP value from the received or port CoS value.
  - In traffic from ingress ports configured with the **trust-cos** port keyword, the **dscp** ACE keyword instructs QoS to set the internal and egress DSCP values from the received or port CoS values.
  - In traffic from ingress ports configured with the **untrusted** port keyword, the **dscp** ACE keyword instructs QoS to set the internal and egress DSCP values from the DSCP value in the ACE.

  Note The default configuration of the ACEs in the default ACLs contains the **dscp** ACE keyword, which supports per-port classification of traffic. With the default values, the ACEs in the default ACLs apply DSCP zero to traffic from ingress ports configured with the **untrusted** port keyword.

QoS uses configurable mapping tables to set the DSCP value, which is 6 bits, from CoS and IP precedence, which are 3-bit values (for more information, see the "Mapping Received CoS Values to Internal DSCP Values" section on page 41-55 and the "Mapping Received IP Precedence Values to Internal DSCP Values" section on page 41-56).

Policing Rules

You can create named policing rules that specify bandwidth utilization limits, which you can apply to traffic by including the policing rule name in an ACE (for more information, see the "Creating Policing Rules" section on page 41-34).

Policing uses a token bucket scheme. As packets arrive, the packet size in bytes is added to the bucket level. Every 0.25 milliseconds, a value equal to the token rate is subtracted from the bucket level.

You specify the bandwidth utilization limits as an average rate and a maximum burst size. Packets that exceed these limits are "out of profile." Traffic is in profile as long as it flows in at an average rate and never bursts beyond the burst size.

In each policing rule, you specify if out-of-profile packets are to be dropped or to have a new DSCP value applied to them (applying a new DSCP value is called "markdown"). Since out-of-profile packets do not retain their original priority, they are not counted as part of the bandwidth consumed by in-profile packets.

For all policing rules, QoS uses a configurable table that maps received DSCP values to marked-down DSCP values (for more information, see the "Mapping DSCP Markdown Values" section on page 41-57). When markdown occurs, QoS gets the marked-down DSCP value from the table. You cannot specify a marked-down DSCP value in individual policing rules.

Note By default, the markdown table is configured so that no markdown occurs: the marked-down DSCP values are equal to the received DSCP values. To enable markdown, configure the table appropriately for your network.

You give each policing rule a unique name when you create it and then use the name to include the policing rule in an ACE. The same policing rule can be used in multiple ACEs.

You can create these policing rules:

- Microflow—QoS applies the bandwidth limit specified in a microflow policing rule separately to each flow that matches any ACEs that use that particular microflow policing rule. You can create up to 63 microflow policing rules.
- Aggregate—QoS applies the bandwidth limits specified in an aggregate policing rule cumulatively to all flows that match any ACEs that use that particular aggregate policing rule. You can create up to 1023 aggregate policing rules.
- With a PFC2, you can specify a dual rate aggregate policing rule with a normal rate and an excess rate.
  - Normal rate—packets exceeding this rate are marked down.
  - Excess rate—packets exceeding this rate are either marked down or dropped as specified by the drop indication flag.

Note The drop indication flag applies to the excess rate policer and cannot be set for the normal rate policer. To achieve the effect of a drop indication flag for the normal rate aggregate policer, set the excess rate equal to the normal rate and set the drop indication flag. Alternatively, you can set the normal rate without specifying an excess rate, which automatically sets the excess rate to the normal rate when the drop indicator flag is on.

You can include both a microflow policing rule and an aggregate policing rule in each ACE to police a flow based on both its own bandwidth utilization and on its bandwidth utilization combined with that of other flows.

For example, you could create a microflow policing rule named “group_individual” with bandwidth limits suitable for individuals in a group and you could create an aggregate policing rule named “group_all” with bandwidth limits suitable for the group as a whole. You could include both policing rules in ACEs that match the group’s traffic. The combination would affect individuals separately and the group cumulatively.

For ACEs that include both a microflow policing rule and an aggregate policing rule, QoS responds to an out-of-profile status from either policing rule and, as specified by the policing rule, applies a new DSCP value or drops the packet. If both policing rules return an out-of-profile status, then if either policing rule specifies that the packet is to be dropped, it is dropped; otherwise, QoS applies a new DSCP value.

Follow these guidelines when creating policing rules:

- You can include a microflow policing rule in IP ACEs. You cannot include a microflow policing rule in IPX or MAC ACEs. IPX and MAC ACEs support only aggregate policing rules.
- By default, microflow policing rules do not affect bridged traffic. To enable microflow policing of bridged traffic, enter the **set qos bridged-microflow-policing** command (for more information, see the “Enabling or Disabling Microflow Policing of Bridged Traffic” section on page 41-48).
- With a Layer 3 Switching Engine II, to do any microflow policing, you must enable microflow policing of bridged traffic.
- With an MSFC, QoS does not apply microflow policing rules to Multilayer Switching (MLS) candidate frames (MSFC2 does not use candidate and enabler frames).
- To avoid inconsistent results, all ACEs that include the same aggregate policing rule must use the same ACE keyword: **trust-dscp**, **trust-ipprec**, **trust-cos**, or **dscp**. If the ACE uses the **dscp** keyword, all traffic that matches the ACE must come through ports configured with the same port keyword: **trust-dscp**, **trust-ipprec**, **trust-cos**, or **untrusted**. If the ACL is attached to a VLAN, all ports in the VLAN must be configured with the same port keyword.

PFC2 Policing Decisions

With a PFC2, the policing decision consists of two levels:

- Normal Police Level—Set if either the microflow policer or the aggregate normal rate policer returns an out-of-profile decision.
- Excess Police Level—Set if the aggregate excess rate policer returns an out-of-profile decision.

Packets are dropped if the excess rate aggregate policer returns an out-of-profile decision and the drop indication flag is set, or if the microflow policer returns an out-of-profile decision and the drop indication flag is set.

If an excess police level is set, the excess DSCP mapping is used to replace the original DSCP value with a marked-down value. If only a normal police level is set, the normal DSCP mapping is used. The excess police level has precedence for selecting mapping rules when both police levels are set because the excess police level represents the worst out-of-profile transgression.

Attaching ACLs

You can configure each port for either port-based QoS (default) or VLAN-based QoS (see the “Enabling Port-Based or VLAN-Based QoS” section on page 41-32) and attach ACLs to the selected interface (see the “Attaching ACLs to Interfaces” section on page 41-46). You can attach up to three named ACLs, one of each type (IP, IPX, and Ethernet) to each port and VLAN.

On ports configured for VLAN-based QoS, you can attach named ACLs to the port's VLAN; or for a trunk, you can attach named ACLs to any VLANs allowed on the trunk as follows:

- On a port configured for VLAN-based QoS, traffic received through the port is compared to any named ACLs attached to the port's VLAN. If you do not attach any named ACLs to the port's VLAN, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic received through the port to the default ACLs.
- On a trunk configured for VLAN-based QoS, traffic received through the port is compared to any named ACLs attached to the traffic's VLAN. For traffic in VLANs that have no named ACLs attached, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic to the default ACLs.

On ports configured for port-based QoS, you can attach named ACLs to the port as follows:

- On a port configured for port-based QoS, traffic received through the port is compared to any named ACLs attached to the port. If you do not attach any named ACLs to the port, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic received through the port to the default ACLs.
- On a trunk configured for port-based QoS, traffic in all VLANs received through the port is compared to any named ACLs attached to the port. If you do not attach any named ACLs to the port, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic received through the port to the default ACLs.

Final Layer 3 Switching Engine CoS and ToS Values

With a Layer 3 switching engine, QoS associates CoS and ToS values with traffic as specified by the marking and policing rules in the ACE that the traffic matches (see the "Internal DSCP Values" section on page 41-15). The associated CoS and ToS are used at the Ethernet egress port (see the "Ethernet Egress Port Scheduling, Congestion Avoidance, and Marking" section on page 41-24).

Classification and Marking with a Layer 2 Switching Engine

With a Layer 2 Switching Engine, QoS can classify traffic addressed to specified MAC address/VLAN pairs to be marked with a configured CoS value (for more information, see the "Definitions" section on page 41-2 and the "Mapping a CoS Value to a Host Destination MAC Address/VLAN Pair" section on page 41-47).

Note Classification and marking with a Layer 2 Switching Engine uses Layer 2 CoS values. Classification and marking with a Layer 2 Switching Engine does not use or set Layer 3 IP precedence or DSCP values.

Ethernet Egress Port Scheduling, Congestion Avoidance, and Marking

These sections describe Ethernet egress port scheduling, congestion avoidance, and marking:

Overview

QoS schedules traffic through the transmit queues based on CoS values and uses CoS-value-based transmit-queue drop thresholds to avoid congestion in traffic transmitted from Ethernet ports.

Note Ethernet egress port scheduling and congestion avoidance uses Layer 2 CoS values. Ethernet egress port marking writes Layer 2 CoS values and, for IP traffic, the Layer 3 ToS byte.

Transmit Queues

Enter the **show port capabilities** command to see the queue structure of a port. The command displays one of the following:

- **tx-(2q2t)**—Two standard queues with two thresholds each
- **tx-(1p2q2t)**—One strict-priority queue and two standard queues with two thresholds each
- **tx-(1p3q1t)**—One strict-priority queue and three standard queues with one threshold each

All ports have a low-priority and a high-priority standard transmit queue. **1p3q1t** ports have a medium-priority standard transmit queue. **1p2q2t** and **1p3q1t** ports have a strict-priority transmit queue in addition to the standard queues.

On **2q2t** ports, the default QoS configuration allocates a minimum of 80 percent of the total transmit queue size to the low-priority standard queue and a minimum of 20 percent to the high-priority standard queue.

On **1p2q2t** and **1p3q1t** ports, the switch services traffic in the strict-priority queue before servicing the standard queues. When the switch is servicing a standard queue, after transmitting a packet, it checks for traffic in the strict-priority queue. If the switch detects traffic in the strict-priority queue, it suspends its service of the standard queue and completes service of all traffic in the strict-priority queue before returning to the standard queue.

On **1p2q2t** ports, the default QoS configuration allocates a minimum of 70 percent of the total transmit queue size to the low-priority standard queue, a minimum of 15 percent to the high-priority standard queue, and a minimum of 15 percent to the strict-priority queue.

On **1p3q1t** ports, the transmit queue size is not configurable and is allocated equally among all queues.

Scheduling and Congestion Avoidance

These sections describe scheduling and congestion avoidance:

- 2q2t Ports, page 41-26
- 1p2q2t Ports, page 41-26
- 1p3q1t Ports, page 41-26

Note The explanations in these sections use default values. You can configure many of the parameters (for more information, see the "Configuring QoS" section on page 41-30). All ports of the same type use the same drop-threshold configuration.

2q2t Ports

For **2q2t** ports, each transmit queue has two drop thresholds that function as follows:

- Frames with CoS 0, 1, 2, or 3 go to the low-priority transmit queue (queue 1):
  - Using transmit queue 1, drop-threshold 1, the switch drops frames with CoS 0 or 1 when the low-priority transmit-queue buffer is 80 percent full.
  - Using transmit queue 1, drop threshold 2, the switch drops frames with CoS 2 or 3 when the low-priority transmit-queue buffer is 100 percent full.
- Frames with CoS 4, 5, 6, or 7 go to the high-priority transmit queue (queue 2):
  - Using transmit queue 2, drop threshold 1, the switch drops frames with CoS 4 or 5 when the high-priority transmit-queue buffer is 80 percent full.
  - Using transmit queue 2, drop threshold 2, the switch drops frames with CoS 6 or 7 when the high-priority transmit-queue buffer is 100 percent full.

1p2q2t Ports

For **1p2q2t** ports, the low- and high-priority standard transmit queues each have two drop thresholds that function as follows:

- Frames with CoS 0, 1, 2, or 3 go to the low-priority standard transmit queue (queue 1):
  - Using standard transmit queue 1, drop threshold 1, the switch drops frames with CoS 0 or 1 when the low-priority transmit-queue buffer is 80 percent full.
  - Using standard transmit queue 1, drop threshold 2, the switch drops frames with CoS 2 or 3 when the low-priority transmit-queue buffer is 100 percent full.
- Frames with CoS 4, 6, or 7 go to the high-priority standard transmit queue (queue 2):
  - Using standard transmit queue 2, drop threshold 1, the switch drops frames with CoS 4 when the high-priority transmit-queue buffer is 80 percent full.
  - Using standard transmit queue 2, drop threshold 2, the switch drops frames with CoS 6 or 7 when the high-priority transmit-queue buffer is 100 percent full.
- Frames with CoS 5 go to the strict-priority transmit queue (queue 3), where the switch drops frames only when the buffer is 100 percent full.

1p3q1t Ports

For **1p3q1t** ports, the queues each have one drop threshold that function as follows:

- Frames with CoS 0 and 1 go to the low-priority standard transmit queue (queue 1).
- Frames with CoS 2, 3, or 4 go to the medium-priority standard transmit queue (queue 2).

- Frames with CoS 6 or 7 go to the high-priority standard transmit queue (queue 3).

Note You can configure each standard transmit queue to use both a tail-drop and a WRED-drop threshold by mapping a CoS value to a queue or to a queue and a threshold. The switch uses tail-drop thresholds for traffic carrying CoS values mapped only to a queue. The switch uses WRED-drop thresholds for traffic carrying CoS values mapped to a queue and a threshold.

- Frames with CoS 5 go to the strict-priority transmit queue (queue 3), where the switch drops frames only when the buffer is 100 percent full.

Marking

When traffic is transmitted from the switch, QoS writes the ToS byte into IP traffic (Layer 3 switching engine only) and the CoS value that was used for scheduling and congestion avoidance into ISL or 802.1Q traffic (for more information, see the "Final Layer 3 Switching Engine CoS and ToS Values" section on page 41-24).

QoS Statistics Data Export

The QoS statistics data export feature generates per port and per aggregate policer utilization information and forwards this information in UDP packets to traffic monitoring, planning, or accounting applications. You can enable QoS statistics data export on a per port or on a per-aggregate policer basis. The statistics data generated per port consists of counts of the input and output packets and bytes. The aggregate policer statistics consists of counts of allowed packets and counts of packets exceeding the policed rate.

The QoS statistics data collection occurs periodically at a fixed interval of 5 minutes, but the interval at which the data is exported is configurable. QoS statistics collection is enabled by default, and the data export feature is disabled by default for all ports and all aggregate policers configured on the Catalyst 6000 family switch.

Note Per-port counter information and utilization statistics are not available for ATM ports.

Note The QoS statistics data export feature is completely separate from TopN and NetFlow Data Export and does not interact with either of these features.

QoS Default Configuration

Table 41-3 shows the QoS default configuration.

Table 41-3 QoS Default Configuration

Feature	Default Value
QoS enable state	Disabled **Note**—With QoS enabled and all other QoS parameters at default values, QoS sets Layer 3 DSCP to zero and Layer 2 CoS to zero in all traffic transmitted from the switch.
Port CoS value	0
IntraVLAN microflow policing	Disabled
CoS to internal DSCP map (internal DSCP set from CoS values)	CoS 0 = DSCP 0 CoS 1 = DSCP 8 CoS 2 = DSCP 16 CoS 3 = DSCP 24 CoS 4 = DSCP 32 CoS 5 = DSCP 40 CoS 6 = DSCP 48 CoS 7 = DSCP 56
IP precedence to internal DSCP map (internal DSCP set from IP precedence values)	IP precedence 0 = DSCP 0 IP precedence 1 = DSCP 8 IP precedence 2 = DSCP 16 IP precedence 3 = DSCP 24 IP precedence 4 = DSCP 32 IP precedence 5 = DSCP 40 IP precedence 6 = DSCP 48 IP precedence 7 = DSCP 56
Internal DSCP to egress CoS map (egress CoS set from internal DSCP values)	DSCP 0–7 = CoS 0 DSCP 8–15 = CoS 1 DSCP 16–23 = CoS 2 DSCP 24–31 = CoS 3 DSCP 32–39 = CoS 4 DSCP 40–47 = CoS 5 DSCP 48–55 = CoS 6 DSCP 56–63 = CoS 7
Marked-down DSCP from DSCP map	Marked-down DSCP value equals original DSCP value (no markdown)
Policing rules	None
Named ACLs	None
Default ACLs	Supports per-port classification and marking, sets DSCP to 0 in traffic from untrusted ports, no policing
COPS[1] support	Disabled
RSVP support	Disabled
QoS statistics data export	Disabled

Table 41-3 QoS Default Configuration (continued)

Feature	Default Value
With QoS enabled	
Runtime—Port based or VLAN based	Port based
Config—Port based or VLAN based	Port based
Port trust state	Untrusted
Receive-queue tail-drop threshold[2] percentages	• Threshold 1: 50% • Threshold 2: 60% • Threshold 3: 80% • Threshold 4: 100%
Transmit-queue tail-drop threshold percentages	• Low-priority queue threshold 1: 80% • Low-priority queue threshold 2: 100% • High-priority queue threshold 1: 80% • High-priority queue threshold 2: 100%
1p2q2t transmit-queue WRED-drop threshold percentages	• Low-priority queue threshold 1: – Low WRED-drop threshold: 40% – High WRED-drop threshold: 70% • Low-priority queue threshold 2: – Low WRED-drop threshold: 70% – High WRED-drop threshold: 100% • High-priority queue threshold 1: – Low WRED-drop threshold: 40% – High WRED-drop threshold: 70% • High-priority queue threshold 2: – Low WRED-drop threshold: 70% – High WRED-drop threshold: 100%
1p3q1t transmit-queue WRED-drop threshold percentages	• Low WRED-drop threshold: 70% • High WRED-drop threshold: 100%
Transmit-queue low-priority/high-priority ratio	4:255
Standard transmit-queue size ratio	• Low priority: 80% • High priority: 20%

Table 41-3 QoS Default Configuration (continued)

Feature	Default Value
CoS value/drop-threshold mapping	• **1q4t**/**2q2t** and **1p1q4t**/**1p2q2t** ports: – Receive queue 1/drop threshold 1 and transmit queue 1/drop threshold 1: CoS 0 and 1 – Receive queue 1/drop threshold 2 and transmit queue 1/drop threshold 2: CoS 2 and 3 – Receive queue 1/drop threshold 3 and transmit queue 2/drop threshold 1: CoS 4 and 5[3] – Receive queue 1/drop threshold 4 and transmit queue 2/drop threshold 2: CoS 6 and 7 • **1p1q0t**/**1p3q1t** ports: – Receive queue 1 (standard) tail-drop threshold: CoS 0, 1, 2, 3, 4, 6, and 7 – Receive queue 2 (priority): CoS 5
With QoS disabled	
Runtime—Port based or VLAN based	VLAN based
Config—Port based or VLAN based	Port based
Port trust state	**trust-cos** (Layer 2 switching engine) **trust-dscp** (Layer 3 switching engine)
Receive-queue drop-threshold percentages	All thresholds set to 100%
Transmit-queue drop-threshold percentages	All thresholds set to 100%
Transmit-queue low-priority/high-priority bandwidth allocation ratio	255:1
Transmit-queue size ratio	• Low priority: 100% • High priority: Not used
CoS value/drop-threshold mapping	Receive-drop threshold 1 and transmit-queue 1/drop threshold 1: CoS 0–7

1. COPS=Common Open Policy Service
2. QoS implements receive-queue drop thresholds only on ports configured with the **trust-cos** port keyword.
3. On **1p1q4t** and **1p2q2t** ports, QoS maps CoS 5 to the strict-priority queues.

Configuring QoS

These sections describe how to configure QoS on the Catalyst 6000 family switches:

Note Some QoS **show** commands support the **config** and **runtime** keywords. Use the **runtime** keyword to display the QoS values currently programmed into the hardware. When you disable QoS, the display with the **runtime** keyword is "QoS is disabled." Use the **config** keyword to display values from commands that have been entered, but which may not currently be programmed into the hardware (for example, locally configured QoS values that are currently not used because COPS has been selected as the QoS policy source or QoS values configured when QoS is disabled).

Enabling QoS

To enable QoS, perform this task in privileged mode:

Task	Command
Enable QoS on the switch.	**set qos** {**enable** | **disable**}

This example shows how to enable QoS:

```
Console> (enable) set qos enable
QoS is enabled.
Console> (enable)
```

Enabling Port-Based or VLAN-Based QoS

Note The commands in this section are not supported with a Layer 2 Switching Engine.

By default, QoS uses ACLs attached to ports. On a per-port basis, you can configure QoS to use ACLs attached to a VLAN. To enable VLAN-based QoS on a port, perform this task in privileged mode:

	Task	Command
Step 1	Enable VLAN-based QoS on a port.	**set port qos** *mod/port* {**port-based** | **vlan-based**}
Step 2	Verify the configuration.	**show port qos** *mod/port*

For more information, see the "Attaching ACLs" section on page 41-23.

This example shows how to enable VLAN-based QoS on a port:

```
Console> (enable) set port qos 1/1-2 vlan-based
Hardware programming in progress...
QoS interface is set to vlan-based for ports 1/1-2.
Console> (enable)
```

Changing a port from port-based to VLAN-based QoS detaches all ACLs from the port. Any ACLs attached to the VLAN apply to the port immediately (for more information, see the "Attaching ACLs to Interfaces" section on page 41-46).

Configuring the Trust State of a Port

This command configures the trust state of a port (for more information, see the "Ethernet Ingress Port Marking, Scheduling, Congestion Avoidance, and Classification" section on page 41-10). By default, all ports are untrusted.

To configure the trust state of a port, perform this task in privileged mode:

	Task	Command
Step 1	Configure the trust state of a port.	**set port qos trust** {**untrusted** | **trust-cos** | **trust-ipprec** | **trust-dscp**}
Step 2	Verify the configuration.	**show port qos**

The **trust-ipprec** and **trust-dscp** keywords are supported only with a Layer 3 switching engine and are not supported on **1q4t** ports except Gigabit Ethernet. On **1q4t** ports (except Gigabit Ethernet), the **trust-cos** port keyword displays an error message, activates receive-queue drop thresholds, and—as indicated by the error message—does not apply the **trust-cos** trust state to traffic. You must configure the **trust-cos** ACL that matches the ingress traffic to apply the **trust-cos** trust state.

This example shows how to configure port 1/1 with the **trust-cos** keyword:

```
Console> (enable) set port qos 1/1 trust trust-cos
Port 1/1 qos set to trust-cos
Console> (enable)
```

Note Only ISL or 802.1Q frames carry CoS values. Configure ports with the **trust-cos** keyword only when the received traffic is ISL or 802.1Q frames carrying CoS values that you know to be consistent with network policy or to trust a configured port CoS value.

Configuring the CoS Value for a Port

Note Whether or not QoS uses the CoS value applied with the **set port qos** command depends on the trust state of the port and the trust state of the traffic received through the port. The **set port qos** command does not configure the trust state of the port or the trust state of the traffic received through the port. To use the CoS value applied with the **set port qos** command, configure the ingress port as trusted or configure a trust-CoS ACL that matches the ingress traffic.

Unmarked frames from ports configured as trusted and all frames from ports configured as untrusted are assigned the CoS value specified with this command.

To configure the CoS value for a port, perform this task in privileged mode:

	Task	Command
Step 1	Configure the CoS value for a port.	**set port qos cos** *cos-value*
Step 2	Verify the configuration.	**show port qos**

This example shows how to configure the port CoS value to 3 for port 1/1:

```
Console> (enable) set port qos 1/1 cos 3
Port 1/1 qos cos set to 3
Console> (enable)
```

To revert to the default CoS value for a port, perform this task in privileged mode:

	Task	Command
Step 1	Revert to the default CoS value for a port.	**clear port qos cos**
Step 2	Verify the configuration.	**show port qos**

This example shows how to revert to the default CoS value for port 1/1:

```
Console> (enable) clear port qos 1/1 cos
Port 1/1 qos cos setting cleared.
Console> (enable)
```

Creating Policing Rules

Note The commands in this section are not supported with a Layer 2 Switching Engine.

To create a policing rule, perform this task in privileged mode:

	Task	Command
Step 1	Create a policing rule.	**set qos policer microflow** *microflow_name* {**rate** *rate*} {**burst** *burst*} {**drop** | **policed-dscp**} With PFC or PFC2: **set qos policer aggregate** *aggregate_name* {**rate** *rate*} {**burst** *burst*} {**drop** | **policed-dscp**} With PFC2: **set qos policer aggregate** *aggregate_name* {**rate** *rate*} **policed-dscp** {**erate** *erate*} {**drop** | **policed-dscp**} **burst** *burst*
Step 2	Verify the configuration.	**show qos policer** {**config** | **runtime**} {**microflow** | **aggregate** | **all**}

For more information, see the "Policing Rules" section on page 41-22.

The *policer_name* parameter can be up to 31 characters long, is case sensitive, and may include a–z, A–Z, 0–9, the dash character (-), the underscore character (_), and the period character (.). Policing rule names must start with an alphabetic character (not a digit) and must be unique across all microflow and aggregate policing rules. You cannot use keywords from any command as a policing rule name.

The valid values for the *rate* and *erate* parameters are 32 Kbps (entered as 32) to 8 Gbps (entered as 8000000); or to classify all traffic as out of profile, set the *rate* parameter to zero (0). The PFC1 and PFC2 have the following hardware granularity for rate values:

Rate Value Range	Granularity	Rate Value Range	Granularity
1 to 1024 (1 Mbs)	32768 (32 K)	65537 to 131072 (128 Mbs)	4194304 (4 M)
1025 to 2048 (2 Mbs)	65536 (64 K)	131073 to 262144 (256 Mbs)	8388608 (8 M)
2049 to 4096 (4 Mbs)	131072 (128 K)	262145 to 524288 (512 Mbs)	16777216 (16 M)
4097 to 8192 (8 Mbs)	262144 (256 K)	524289 to 1048576 (1 Gps)	33554432 (32 M)
8193 to 16384 (16 Mbs)	524288 (512 K)	1048577 to 2097152 (2 Gps)	67108864 (64 M)
16385 to 32768 (32 Mbs)	1048576 (1 M)	2097153 to 4194304 (4 Gps)	134217728 (128 M)
32769 to 65536 (64 Mbs)	2097152 (2 M)	4194305 to 8000000 (8 Gps)	268435456 (256 M)

Within each range, QoS programs the hardware with rate values that are multiples of the granularity values.

The valid values for the *burst* parameter are 1 Kb (entered as 1) to 32 Mb (entered as 32000).

Note The *burst* parameter sets the token bucket size. To sustain a specific rate, set the token bucket size with the *burst* parameter to be at least the *rate* divided by 4000, because tokens are removed from the bucket every 1/4000th of a second (0.25 ms) and the bucket needs to be at least *burst*-size long to sustain the specified rate.

Note Because any packet larger than the burst size is considered an out-of-profile packet, make sure that the burst size is greater than or equal to the largest packet size of the policer that is applied to it.

Note QoS programs the hardware with values that are multiples of 32K (32,768), not with the specific value entered.

Enter either the **drop** keyword to cause all out-of-profile packets to be dropped or the **policed-dscp** keyword to cause all out-of-profile packets with the normal rate to be marked down as specified in the normal markdown DSCP map (for more information, see the "Mapping DSCP Markdown Values" section on page 41-57).

This example shows how to create a microflow policing rule with a 1-Mbps rate limit and a 10-Mb burst limit that marks down out-of-profile traffic:

```
Console> (enable) set qos policer microflow my-micro rate 1000 burst 10000
policed-dscp
Hardware programming in progress...
QoS policer for microflow my-micro created successfully.
Console> (enable)
```

For PFC2, this example shows how to create an aggregate excess rate policing rule with a 64-Kbps rate limit and a 128-Kb burst limit that drops traffic exceeding these values:

```
Console> (enable) set qos policer aggregate test rate 64 burst 128 drop
QoS policer for aggregate test created successfully.
Console> (enable) show qos policer config aggregate test
QoS aggregate policers:
QoS aggregate policers:
Aggregate name                Normal rate (kbps) Burst size (kb) Normal action
----------------------------- ------------------ --------------- -------------
test                                          64             128    policed-dscp
                              Excess rate (kbps) Burst size (kb) Excess action
                              ------------------ --------------- -------------
                                              64             128    drop
                              ACL attached
 -----------------------------------
Console> (enable)
```

For PFC2, this example shows how to create an aggregate excess rate policing rule with a 64-Kbps rate limit and a 100-Kb burst limit that will cause all out-of-profile packets to be marked down as specified in the normal markdown DSCP map:

```
Console> (enable) set qos policer aggregate test2 rate 64 burst 100 policed-dscp
QoS policer for aggregate test2 created successfully.
```

```
Console> (enable) show qos policer config aggregate test2
QoS aggregate policers:
Aggregate name                Normal rate (kbps)  Burst size (kb) Normal action
----------------------------- ------------------  --------------- -------------
test2                                     64             100  policed-dscp
                              Excess rate (kbps)  Burst size (kb) Excess action
                              ------------------  --------------- ---------------
                                        8000000             100  policed-dscp
                              ACL attached
                              ------------------------------------

Console> (enable)
```

For PFC2, this example shows how to create an aggregate excess rate policing rule with a 64-Kbps rate limit and a 128-Kb burst limit that will cause traffic that exceeds the normal rate of 64 Kbps and a burst size of 96 Kb to be marked down as specified in the normal markdown DSCP map, and traffic that exceeds 128 Kbps and a burst size of 96 Kb to be dropped:

```
Console> (enable) set qos policer aggregate test3 rate 64 policed-dscp erate 128 drop
burst 96
QoS policer for aggregate test3 created successfully.
Console> (enable) show qos policer config aggregate test3
QoS aggregate policers:
Aggregate name                Normal rate (kbps)  Burst size (kb) Normal action
----------------------------- ------------------  --------------- -------------
test3                                     64              96 policed-dscp
                              Excess rate (kbps)  Burst size (kb) Excess action
                              ------------------  --------------- ---------------
                                           128              96 drop
                              ACL attached
                              ------------------------------------

Console> (enable)
```

Deleting Policing Rules

Note You can only delete policing rules if they are not attached to any interfaces (for more information, see the "Detaching ACLs from Interfaces" section on page 41-46).

To delete one or all policing rules, perform this task in privileged mode:

	Task	Command
Step 1	Delete one or all policing rules.	**clear qos policer** {**microflow** | **aggregate**} {*policer_name* | **all**}
Step 2	Verify the configuration.	**show qos policer** {**config** | **runtime**} {**microflow** | **aggregate** | **all**}

This example shows how to delete the microflow policing rule named my_micro:

```
Console> (enable) clear qos policer microflow my_micro
my_micro QoS microflow policer cleared.
Console> (enable)
```

Creating or Modifying ACLs

Note The commands in this section are not supported with a Layer 2 Switching Engine.

These sections describe ACL creation and modification:

ACL Names

ACL names can be up to 31 characters long, are case sensitive, and may include a–z, A–Z, 0–9, the dash character (-), the underscore character (_), and the period character (.). ACL names must start with an alphabetic character and must be unique across all QoS ACLs of all types. You cannot use keywords from any command as an ACL name.

ACE Name, Marking Rule, Policing, and Filtering Syntax

ACE command syntax is organized as follows:
ACL_command ACL_type_and_name marking_rule policing_rule filtering

For example, in an IP ACE, the command syntax is as follows:

set qos acl ip *acl_name* {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] *src_ip_spec* [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]

- **set qos acl ip** *acl_name*—Creates a named ACL of the specified type or adds the ACE to the ACL if it already exists. See the "ACL Names" section on page 41-37.
- {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**}—Selects a marking rule. See the "Marking Rules" section on page 41-21.
- [**microflow** *microflow_name*] [**aggregate** *aggregate_name*]—Optionally configures policing in the ACE. See the "Policing Rules" section on page 41-22.
- *src_ip_spec* [**precedence** *precedence* | **dscp-field** *dscp*]—The rest of the parameters, except the **editbuffer** keywords, configure filtering.

Named IP ACLs

These sections describe creating or modifying IP ACLs:

Source and Destination IP Addresses and Masks

In IP ACEs, specify source and destination IP addresses and masks (represented by the *src_ip_spec* and *dest_ip_spec* parameters in the following sections) in the form *ip_address mask*. The mask is mandatory. Use one bits, which need not be contiguous, where you want wildcards.

Use any of the following formats for the address and mask:

- Four-part dotted-decimal 32-bit values
- The keyword **any** as an abbreviation for a wildcard address and wildcard mask of 0.0.0.0 255.255.255.255
- The abbreviation **host** *ip_address* for an address and wildcard mask of *ip_address* 0.0.0.0

Port Operator Parameters

In IP ACEs, the *operator* parameter can be one of the following:

- **lt** (less than)
- **gt** (greater than)
- **eq** (equal)
- **neq** (not equal)
- **range** (with a pair of port parameters)

See the "Guidelines for Using Layer 4 Operations" section on page 16-19 for restrictions that apply to QoS ACLs.

Precedence Parameter Options

For *precedence* parameter keyword options in IP ACEs, see the "IP ACE Layer 3 Classification Criteria" section on page 41-16.

IP ACEs for TCP Traffic

To create or modify an IP ACE for TCP traffic, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IP ACE for TCP traffic.	**set qos acl ip** {*acl_name*} {{**dscp** *dscp*} | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] **tcp** {*src_ip_spec*} [{*operator*} {*port*} [*port*]] {*dest_ip_spec*} [{*operator*} {*port*} [*port*]] [**established**] [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

For *port* parameter keyword options, see the "IP ACE Layer 4 TCP Classification Criteria" section on page 41-17.

The **established** keyword matches traffic with the ACK or RST bits set.

This example shows how to create an IP ACE for TCP traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro aggregate my-agg
tcp any any
my_IPacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

IP ACEs for UDP Traffic

To create or modify an IP ACE for UDP traffic, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IP ACE for UDP traffic.	**set qos acl ip** {*acl_name*} {{**dscp** *dscp*} | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] **udp** {*src_ip_spec*} [{*operator*} {*port*} [*port*]] {*dest_ip_spec*} [{*operator*} {*port*} [*port*]] [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

For *port* parameter keyword options, see the "IP ACE Layer 4 UDP Classification Criteria" section on page 41-18.

This example shows how to create an IP ACE for UDP traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro aggregate my-agg
udp any any
my_IPacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

IP ACEs for ICMP Traffic

To create or modify an IP ACE for ICMP traffic, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IP ACE for ICMP traffic.	**set qos acl ip** *acl_name* {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] **icmp** *src_ip_spec dest_ip_spec* [*icmp_type* [*icmp_code*] | *icmp_message*] [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

For *icmp_code* and *icmp_type* parameter keyword options, see the "IP ACE Layer 4 ICMP Classification Criteria" section on page 41-18.

This example shows how to create an IP ACE for ICMP *echo* traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro aggregate my-agg
icmp any any echo
my_IPacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

IP ACEs for IGMP Traffic

Note QoS does not support IGMP traffic when IGMP snooping is enabled.

To create or modify an IP ACE for IGMP traffic, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IP ACE for IGMP traffic.	**set qos acl ip** *acl_name* {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] **igmp** *src_ip_spec dest_ip_spec* [*igmp_type*] [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

For *igmp_type* parameter keyword options, see the "IP ACE Layer 4 IGMP Classification Criteria" section on page 41-19.

This example shows how to create an IP ACE for IGMP protocol independent multicast (PIM) traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro aggregate my-agg
igmp any any pim
my_IPacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

IP ACLs for Other Layer 4 Protocols

To create or modify a named IP ACL with additional parameters that match all Layer 4 protocols, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IP ACE.	**set qos acl ip** *acl_name* {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] *protocol src_ip_spec dest_ip_spec* [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

For *protocol* parameter keyword options, see the "IP ACE Layer 4 Protocol Classification Criteria" section on page 41-17.

This example shows how to create an IP ACE for IPINIP traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro aggregate my-agg
ipinip any any
my_IPacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

IP ACEs for Any IP Traffic

To create or modify an IP ACE that matches all IP traffic, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IP ACE.	**set qos acl ip** *acl_name* {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] *src_ip_spec* [**precedence** *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

This example shows how to create an IP ACE:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro aggregate my-agg
any
my_IPacl editbuffer modified. Use ‘commit’ command to apply changes.
Console> (enable)
```

Modifying the Default IP ACL

To modify the default IP ACL, perform this task in privileged mode:

	Task	Command
Step 1	Modify the default IP ACL.	**set qos acl default-action ip** {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*]
Step 2	Verify the configuration.	**show qos acl info default-action** {**ip** | **ipx** | **mac** | **all**}

For more information, see the “Default ACLs” section on page 41-20.

This example shows how to modify the default IP ACL:

```
Console> (enable) set qos acl default-action ip dscp 5 microflow my-micro aggregate my-agg
QoS default-action for IP ACL is set successfully.
Console> (enable)
```

Creating or Modifying Named IPX ACLs

To create or modify a named IPX ACL, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify an IPX ACL.	With PFC: **set qos acl ipx** *acl_name* {**dscp** *dscp* | **trust-cos**} [**aggregate** *aggregate_name*] *protocol src_net* [*dest_net*[.*dest_node*] [[*dest_net_mask*].*dest_node_mask*]] [**before** *editbuffer_index* | **modify** *editbuffer_index*] With PFC2: **set qos acl ipx** *acl_name* **aggregate** *aggregate_name protocol src_net* [*dest_net*[.*dest_node*] [[*dest_net_mask*].*dest_node_mask*]] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

The *protocol* parameter can be specified numerically (0–255) or with these keywords: **any**, **ncp** (17), **netbios** (20), **rip** (1), **sap** (4), or **spx** (5).

The *src_net* and *dest_net* parameters are IPX network numbers, entered as up to 8 hexadecimal digits in the range 1 to FFFFFFFE (-1 matches any network number). You do not need to enter leading zeros.

If you specify an IPX destination network, IPX ACEs support the following optional parameters:

- An IPX destination network mask, entered as up to 8 hexadecimal digits in the range 1 to FFFFFFFE (-1 matches any network number). Use one bits, which need not be contiguous, where you want wildcards.
- An IPX destination node, entered as 12 hexadecimal digits (48 bits), formatted as a dotted triplet of four-digit hexadecimal digits each (xxxx.xxxx.xxxx).
- If you specify an IPX destination node, IPX ACEs support an IPX destination node mask, entered as 12 hexadecimal digits (48 bits), formatted as a dotted triplet of four-digit hexadecimal digits each (xxxx.xxxx.xxxx). Use one bits, which need not be contiguous, where you want wildcards.

This example shows how to create an IPX ACE:

```
Console> (enable) set qos acl ipx my_IPXacl trust-cos aggregate my-agg -1
my_IPXacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

Creating or Modifying Named MAC ACLs

To create or modify a named MAC ACL, perform this task in privileged mode:

	Task	Command
Step 1	Create or modify a MAC ACL.	With PFC: **set qos acl mac** *acl_name* {**dscp** *dscp* | **trust-cos**} [**aggregate** *aggregate_name*] *src_mac_spec dest_mac_spec* [*ethertype*] [**before** *editbuffer_index* | **modify** *editbuffer_index*] With PFC2: **set qos acl mac** *acl_name* **aggregate** *aggregate_name src_mac_spec dest_mac_spec* [*ethertype*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {**acl_name** | **all**} **editbuffer** [*editbuffer_index*]

Enter the *src_mac_spec* and *dest_mac_spec* parameters as a MAC address and a mask. Each parameter is 12 hexadecimal digits (48 bits), formatted as dash-separated pairs. Use one bits, which need not be contiguous, where you want wildcards. Use the **any** keyword for a MAC address and mask of 0-0-0-0-0-0 ff-ff-ff-ff-ff-ff. Use the **host** keyword with a MAC address to specify an all-zero mask (*mac_address* 0-0-0-0-0-0).

Enter the *ethertype* parameter as 4 hexadecimal digits (16 bits) prefaced with **0x** (for example, 0x0600) or as a keyword (see the "MAC ACE Layer 2 Classification Criteria" section on page 41-20).

This example shows how to create a MAC ACE:

```
Console> (enable) set qos acl mac my_MACacl trust-cos aggregate my-agg any any
my_MACacl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

Note QoS MAC ACLs that do not include an ethertype parameter match traffic with any value in the ethertype field, which allows MAC-level QoS to be applied to any traffic except IP and IPX.

Creating or Modifying the Default IPX and MAC ACLs

To create or modify the default IPX or MAC ACL, perform this task in privileged mode:

	Task	Command
Step 1	Modify the default IPX or MAC ACL.	With PFC: **set qos acl default-action** {**ipx** | **mac**} {**dscp** *dscp* | **trust-cos**} [**aggregate** *aggregate_name*] With PFC2: **set qos acl default-action** {**ipx** | **mac**} **aggregate** *aggregate_name*
Step 2	Verify the configuration.	**show qos acl info default-action** {**ip** | **ipx** | **mac** | **all**}

For more information, see the "Default ACLs" section on page 41-20.

This example shows how to modify the default IPX ACL:

```
Console> (enable) set qos acl default-action ipx dscp 5 aggregate my-agg
QoS default-action for IPX ACL is set successfully.
Console> (enable)
```

Note IPX and MAC ACLs do not support microflow policing rules.

Deleting Named ACLs

To delete a named ACL, perform this task in privileged mode:

	Task	Command
Step 1	Delete a named ACL.	**clear qos acl** *acl_name* [*editbuffer_index*]
Step 2	Verify the configuration.	**show qos acl info** {*acl_name* | **all**}

This example shows how to delete the ACL named icmp_acl:

```
Console> (enable) clear qos acl icmp_acl 1
ACL icmp_acl ACE# 1 is deleted.
icmp_acl editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
```

Reverting to Default Values in Default ACLs

To revert to the default values for a default ACL, perform this task in privileged mode:

	Task	Command
Step 1	Revert to the default values for a default ACL.	**clear qos acl default-action** {**ip** | **ipx** | **mac**}
Step 2	Verify the configuration.	**show qos acl info default-action** {**ip** | **ipx** | **mac** | **all**}

This example shows how to revert to the default values for the default IP ACL:

```
Console> (enable) clear qos acl default-action ip
Hardware programming in progress...
QoS default-action for IP ACL is restored to default setting.
Console> (enable)
```

Discarding Uncommitted ACLs

To discard an uncommitted new ACL or uncommitted changes to an existing ACL, perform this task in privileged mode:

	Task	Command
Step 1	Discard an uncommitted ACL.	**rollback qos acl** {*acl_name* | **all**}
Step 2	If you discarded changes to an existing ACL, verify the configuration.	**show qos acl info** {*acl_name* | **all**}

This example shows how to discard an uncommitted ACL named my_acl:

```
Console> (enable) rollback qos acl my_acl
Rollback for QoS ACL my_acl is successful.
Console> (enable)
```

Note Changes to the default ACLs take effect immediately and cannot be discarded.

Committing ACLs

When you create, change, or delete a named ACL, the changes exist temporarily in an edit buffer in memory. To commit the ACL so that it can be used, perform this task in privileged mode:

	Task	Command
Step 1	Commit an ACL.	**commit qos acl** *acl_name*
Step 2	Verify the configuration.	**show config qos acl** {*acl_name* | **all**}

This example shows how to commit an ACL named my_acl:

```
Console> (enable) commit qos acl my_acl
Hardware programming in progress...
ACL my_acl is committed to hardware.
Console> (enable)
```

Note When you commit an ACL that has already been attached to interfaces, the new values go into effect immediately. Changes to the default ACLs do not need to be committed.

See "Configuring and Storing VACLs and QoS ACLs in Flash Memory" section on page 16-41 for information about where QoS ACLs are stored.

Attaching ACLs to Interfaces

Note The commands in this section are not supported with a Layer 2 Switching Engine.

You can attach one ACL of each type to each VLAN and to each port configured for port-based QoS. You cannot attach ACLs to a port configured for VLAN-based QoS (for more information, see the "Enabling Port-Based or VLAN-Based QoS" section on page 41-32). When an ACL of a particular type (IP, IPX, or Ethernet) is already attached to an interface, attaching a different ACL of the same type detaches the previous ACL.

To attach an ACL to a port or a VLAN, perform this task in privileged mode:

	Task	Command
Step 1	Attach an ACL to an interface.	**set qos acl map** *acl_name* {*mod/port* | *vlan*}
Step 2	Verify the configuration.	**show qos acl map** {**config** | **runtime**} {*acl_name* | *mod/port* | *vlan* | **all**}

This example shows how to attach an ACL named my_acl to port 2/1:

```
Console> (enable) set qos acl map my_acl 2/1
Hardware programming in progress...
ACL my_acl is attached to port 2/1.
Console> (enable)
```

This example shows how to attach an ACL named my_acl to VLAN 4:

```
Console> (enable) set qos acl map my_acl 4
Hardware programming in progress...
ACL my_acl is attached to vlan 4.
Console> (enable)
```

Note The default ACLs do not need to be attached to any interfaces.

Detaching ACLs from Interfaces

Note The commands in this section are not supported with a Layer 2 Switching Engine.

To detach an ACL from a port or a VLAN, perform this task in privileged mode:

	Task	Command
Step 1	Detach an ACL from an interface.	**clear qos acl map** *acl_name* {*mod/port* | *vlan* | **all**}
Step 2	Verify the configuration.	**show qos acl map** {**config** | **runtime**} {*acl_name* | *mod/port* | *vlan* | **all**}

This example shows how to detach an ACL named my_acl from port 2/1:

```
Console> (enable) clear qos acl map my_acl 2/1
Hardware programming in progress...
ACL my_acl is detached from port 2/1.
Console> (enable)
```

This example shows how to detach an ACL named my_acl from VLAN 4:

```
Console> (enable) clear qos acl map my_acl 4
Hardware programming in progress...
ACL my_acl is detached from vlan 4.
Console> (enable)
```

Note The default ACLs cannot be detached from any interfaces.

Mapping a CoS Value to a Host Destination MAC Address/VLAN Pair

Note QoS only supports this command with a Layer 2 Switching Engine.

To map a CoS value to all frames destined for a particular host destination MAC address and VLAN number value pair, perform this task in privileged mode:

	Task	Command
Step 1	Map a CoS value to a host destination MAC address/VLAN pair.	**set qos mac-cos** *dest_mac VLAN cos_value*
Step 2	Verify the configuration.	**show qos mac-cos** {*dest_mac* [*vlan*] | **all**}

This example shows how to map CoS 2 to a destination MAC address and VLAN 525:

```
Console> (enable) set qos mac-cos 00-40-0b-30-03-48 525 2
CoS 2 is assigned to 00-40-0b-30-03-48 vlan 525.
Console> (enable)
```

Deleting a CoS Value to a Host Destination MAC Address/VLAN Pair

Note QoS only supports this command with a Layer 2 Switching Engine.

To delete a host destination MAC address and VLAN number value pair CoS assignment, perform this task in privileged mode:

	Task	Command
Step 1	Delete a host destination MAC address and VLAN number value pair CoS assignment.	**clear qos mac-cos** {*dest_mac* [*vlan*] | **all**}
Step 2	Verify the configuration.	**show qos mac-cos** {*dest_mac* [*vlan*] | **all**}

This example shows how to delete all CoS assignments to destination MAC addresses and VLANs:

```
Console> (enable) clear qos mac-cos all
All CoS to Mac/Vlan entries are cleared.
Console> (enable)
```

Enabling or Disabling Microflow Policing of Bridged Traffic

Note The commands in this section are not supported with a Layer 2 Switching Engine.

By default, microflow policing rules affect only Layer 3-switched traffic. To enable or disable microflow policing of bridged traffic on the switch or on specified VLANs, perform one of these tasks in privileged mode:

Task	Command
• Enable microflow policing of bridged traffic on the switch or on specified VLANs.	**set qos bridged-microflow-policing** {**enable** | **disable**} *vlan*
• Disable microflow policing of bridged traffic on the switch or on specified VLANs.	**set qos bridged-microflow-policing** {**enable** | **disable**} *vlan*
• Verify the configuration.	**show qos bridged-packet-policing** {**config** | **runtime**} *vlan*

Note With Layer 3 Switching Engine II, to do any microflow policing, you must enable microflow policing of bridged traffic.

For more information, see the "Policing Rules" section on page 41-22.

This example shows how to enable microflow policing of traffic in VLANs 1 through 20:

```
Console> (enable) set qos bridged-microflow-policing enable 1-20
QoS microflow policing is enabled for bridged packets on vlans 1-20.
Console> (enable)
```

Configuring Standard Receive-Queue Tail-Drop Thresholds

To configure the standard receive-queue tail-drop thresholds on the switch, perform this task in privileged mode:

Task	Command
Configure the standard receive-queue tail-drop thresholds.	**set qos drop-threshold** *port_type* **rx queue 1** *thr1 thr2 thr3 thr4*

For more information, see the "Receive Queues" section on page 41-11.

QoS maintains separate configurations for **1q4t** ports and **1p1q4t** ports. With either keyword, this command configures only the standard queue. Specify queue 1 for both port types (the threshold in the strict-priority queue is not separately configurable; it uses threshold 4 as specified for queue 1).

The thresholds are all specified as percentages ranging from 1 to 100. A value of 10 indicates a threshold when the buffer is 10 percent full.

This example shows how to configure the standard receive-queue tail-drop thresholds:

```
Console> (enable) set qos drop-threshold 1q4t rx queue 1 20 40 75 100
Receive drop thresholds for queue 1 set at 20% 40% 75% 100%
Console> (enable)
```

Note You cannot configure a drop threshold in a **1p1q0t** receive queue.

Configuring 2q2t Port Standard Transmit-Queue Tail-Drop Thresholds

To configure the standard transmit-queue tail-drop thresholds on all **2q2t** ports, perform this task in privileged mode:

Task	Command
Configure the standard transmit-queue tail-drop thresholds on all **2q2t** ports.	**set qos drop-threshold** *port_type* **tx queue** *q# thr1 thr2*

Queue number 1 is the low-priority transmit queue and queue number 2 is high priority. In each queue, the low-priority threshold number is 1 and the high-priority threshold number is 2.

The thresholds are all specified as percentages ranging from 1 to 100. A value of 10 indicates a threshold when the buffer is 10 percent full.

This example shows how to configure the low-priority transmit-queue tail-drop thresholds:

```
Console> (enable) set qos drop-threshold 2q2t tx queue 1 40 100
Transmit drop thresholds for queue 1 set at 40%  100%
Console> (enable)
```

Note You cannot configure the tail-drop thresholds in **1p3q1t** transmit queues.

Configuring Standard Transmit-Queue WRED-Drop Thresholds

1p2q2t and **1p3q1t** ports have weighted early random detection (WRED)-drop thresholds in their standard transmit queues.

Note **1p3q1t** ports also have nonconfigurable tail-drop thresholds (see the "1p3q1t Ports" section on page 41-26).

To configure the standard transmit-queue WRED-drop thresholds on all ports of each type, perform this task in privileged mode:

Task	Command
Configure the standard transmit-queue WRED-drop thresholds on all ports of a given type.	**set qos wred 1p2q2t** [**tx**] **queue** *q#* [*thr1Lo*:]*thr1Hi* [*thr2Lo*:]*thr2Hi* **set qos wred 1p3q1t** [**tx**] **queue** *q#* [*thr1Lo*:]*thr1Hi*

For **1p2q2t** ports, queue number 1 is the low-priority transmit queue and queue number 2 is high priority. In each queue, the low-priority threshold is number 1 and the high-priority threshold is number 2.

For **1p3q1t** ports, queue number 1 is the low-priority transmit queue, queue number 2 is medium priority, and queue number 3 is high priority. In each queue, the threshold is number 1.

The thresholds are all specified as percentages ranging from 0 to 100. A value of 10 indicates a threshold when the buffer is 10 percent full.

You can configure both the low WRED threshold and the high WRED threshold. You must set the low threshold to a lower percentage than the high threshold.

The low WRED threshold is the traffic level under which no traffic is dropped. The high WRED threshold is the traffic level above which all traffic is dropped. Traffic in the queue between the low and high WRED thresholds has an increasing chance of being dropped as the queue fills. The default low WRED threshold is zero (all traffic has some chance of being dropped).

This example shows how to configure the low-priority transmit-queue WRED-drop thresholds:

```
Console> (enable) set qos wred 1p2q2t queue 1 40:70 70:100
WRED thresholds for queue 1 set to 40:70 and 70:100  on all WRED-capable 1p2q2t ports.
Console> (enable)
```

Note The threshold in the strict-priority queue is not configurable.

Allocating Bandwidth Between Standard Transmit Queues

The switch transmits frames from one standard queue at a time using a weighted-round robin (WRR) algorithm. WRR uses a weight value to decide how much to transmit from one queue before switching to the other. The higher the weight assigned to a queue, the more transmit bandwidth is allocated to it.

To allocate bandwidth between standard transmit queues, perform this task in privileged mode:

Task	Command
Allocate bandwidth between standard transmit queues.	**set qos wrr** *port_type queue1-weight queue2-weight* [*queue3-weight*]

QoS maintains separate configurations for each port type. This command configures only the standard queues; the strict-priority queue requires no configuration. The valid values for weight range from 1–255.

This example shows how to allocate bandwidth for the **2q2t** ports:

```
Console> (enable) set qos wrr 2q2t 30 70
QoS wrr ratio is set successfully.
Console> (enable)
```

Configuring the Receive-Queue Size Ratio

For **1p1q0t** ports, estimate the mix of standard-priority and strict-priority traffic on your network (for example, 85 percent standard-priority traffic and 15 percent strict-priority traffic). Specify queue ratios with the estimated percentages, which must range from 1 to 99 and together add up to 100.

To configure the receive-queue size ratio, perform this task in privileged mode:

Task	Command
Configure the receive-queue size ratio between receive queue 1 (standard priority) and receive queue 2 (strict priority).	**set qos rxq-ratio 1p1q0t** *queue1-val queue2-val*

This example shows how to configure the receive-queue size ratio:

```
Console> (enable) set qos rxq-ratio 1p1q0t 80 20
QoS rxq-ratio is set successfully.
Console> (enable)
```

Configuring the Transmit-Queue Size Ratio

Estimate the mix of traffic of various priorities on your network (for example, 75 percent low-priority traffic, 15 percent high-priority traffic, and 10 percent strict-priority traffic). Specify queue ratios with the estimated percentages, which must range from 1 to 99 and together add up to 100.

To configure the transmit-queue size ratio for each port type, perform this task in privileged mode:

Task	Command
Configure the transmit-queue size ratio.	**set qos txq-ratio** *port_type queue1-val queue2-val* [*queue3-val*]

Valid *port_type* parameters are **2q2t** and **1p2q2t**. QoS maintains separate configurations for each port type. This example shows how to configure the transmit-queue size ratio:

```
Console> (enable) set qos txq-ratio 2q2t 80 20
QoS txq-ratio is set successfully.
Console> (enable)
```

Mapping CoS Values to Drop Thresholds

This command associates CoS values with receive- and transmit-queue drop thresholds. QoS maintains separate configurations for each port type.

These sections describe mapping CoS values to drop thresholds:

Associating 1q4t, 2q2t Ports

To associate CoS values to the drop thresholds on **1q4t**, **2q2t** ports, perform this task in privileged mode:

	Task	Command
Step 1	Associate a CoS value to a drop threshold.	**set qos map 2q2t tx** *q# thr#* **cos** *coslist*
Step 2	Verify the configuration.	**show qos info config** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}

The receive- and transmit-drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2
- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

Use the transmit queue and transmit-queue drop-threshold values in this command. This example shows how to associate the CoS values 0 and 1 to both standard receive-queue 1/threshold 1 and standard transmit- queue 1/threshold 1:

```
Console> (enable) set qos map 2q2t tx 1 1 cos 0,1
Qos tx priority queue and threshold mapped to cos successfully.
Console> (enable)
```

Associating 1p1q4t, 1p2q2t Ports

On **1p1q4t**, **1p2q2t** ports, you configure the receive queues and the transmit queues separately.

1p1q4t Receive Queues

To associate CoS values to **1p1q4t** receive-queue drop thresholds, perform this task in privileged mode:

	Task	Command
Step 1	Associate a CoS value to a receive-queue drop threshold.	**set qos map 1p1q4t rx** *q# thr#* **cos** *coslist*
Step 2	Verify the configuration.	**show qos info config** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}

Queue 1 is the standard queue. Queue 2 is the strict-priority queue.

Threshold numbers range from 1 for low priority to 4 for high priority.

This example shows how to associate the CoS value 5 to strict-priority receive-queue 2/threshold 1:

```
Console> (enable) set qos map 1p1q4t rx 2 1 cos 5
Qos rx strict queue and threshold mapped to cos successfully.
Console> (enable)
```

1p2q2t Transmit Queues

To associate CoS values to the**1p2q2t** transmit-queue drop thresholds, perform this task in privileged mode:

	Task	Command
Step 1	Associate a CoS value to a transmit-queue drop threshold.	**set qos map 1p2q2t tx** *q# thr#* **cos** *coslist*
Step 2	Verify the configuration.	**show qos info config** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}

Queue 1 is standard low priority, queue 2 is high priority, and queue 3 is strict priority.

Threshold 1 low priority and 2 is high priority.

This example shows how to associate the CoS value 5 to strict-priority transmit-queue 3/drop threshold 1:

```
Console> (enable) set qos map 1p2q2t tx 3 1 cos 5
Qos tx strict queue and threshold mapped to cos successfully.
Console> (enable)
```

Associating 1p1q0t, 1p3q1t Ports

On **1p1q0t**, **1p3q1t** ports, you configure the receive queues and the transmit queues separately.

1p1q0t Receive Queues

To associate CoS values to a **1p1q0t** receive queue, perform this task in privileged mode:

	Task	Command
Step 1	Associate a CoS value to a receive queue.	**set qos map 1p1q0t rx** *q#* **cos** *coslist*
Step 2	Verify the configuration.	**show qos info config** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}

Queue 1 is the standard queue. Queue 2 is the strict-priority queue.

This example shows how to associate the CoS value 5 to strict-priority receive-queue 2:

```
Console> (enable) set qos map 1p1q0t rx 2 cos 7
QoS queue mapped to cos successfully.
Console> (enable)
```

1p3q1t Transmit Queues

With **1p3q1t** transmit queues, you can associate a CoS value with either the nonconfigurable tail-drop threshold or the configurable WRED-drop threshold:

- To associate a CoS value with the tail-drop threshold, map the CoS value to the queue.
- To associate a CoS value with the WRED-drop threshold, map the CoS value to the queue and threshold.

To associate CoS values to the **1p3q1t** transmit-queue drop thresholds, perform this task in privileged mode:

	Task	Command
Step 1	Associate a CoS value to a transmit-queue drop threshold.	**set qos map 1p3q1t tx** *q#* [*thr#*] **cos** *coslist*
Step 2	Verify the configuration.	**show qos info config** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}

Queue 1 is standard low priority, queue 2 is medium priority, and queue 3 is high priority. Queue 4 is strict priority.

To map CoS values to the tail-drop threshold, omit the threshold number or enter 0.

The WRED-drop threshold number is 1.

This example shows how to associate the CoS value 0 to transmit-queue 1/drop threshold 1:

```
Console> (enable) set qos map 1p3q1t tx 1 1 cos 0
Qos tx strict queue and threshold mapped to cos successfully.
Console> (enable)
```

Reverting to CoS Map Defaults

To revert to default CoS value/drop threshold mapping, perform this task in privileged mode:

	Task	Command
Step 1	Revert to QoS map defaults.	**clear qos map** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}
Step 2	Verify the configuration.	**show qos info config** {**1p1q4t rx** | **1p1q0t rx** | **1p2q2t tx** | **2q2t tx** | **1p3q1t tx**}

This example shows how to revert to QoS map defaults:

```
Console> (enable) clear qos map 1p3q1t tx
Qos map setting cleared.
Console> (enable)
```

Configuring DSCP Value Maps

Note The commands in this section are not supported with a Layer 2 Switching Engine.

These sections describe how DSCP values are mapped to other values:

Mapping Received CoS Values to Internal DSCP Values

To map received CoS values to the internal DSCP value (see the "Internal DSCP Values" section on page 41-15), perform this task in privileged mode:

	Task	Command
Step 1	Map received CoS values to internal DSCP values.	**set qos cos-dscp-map** *dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8*
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

Enter 8 DSCP values to which QoS maps received CoS values 0 through 7. This example shows how to map received CoS values to internal DSCP values:

```
Console> (enable) set qos cos-dscp-map 20 30 1 43 63 12 13 8
QoS cos-dscp-map set successfully.
Console> (enable)
```

To revert to default CoS to DSCP value mapping, perform this task in privileged mode:

	Task	Command
Step 1	Revert to CoS value/DSCP value map defaults.	**clear qos cos-dscp-map**
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

This example shows how to revert to CoS-DSCP map defaults:

```
Console> (enable) clear qos cos-dscp-map
QoS cos-dscp-map setting restored to default.
Console> (enable)
```

Mapping Received IP Precedence Values to Internal DSCP Values

To map received IP precedence values to the internal DSCP value (see the "Internal DSCP Values" section on page 41-15), perform this task in privileged mode:

	Task	Command
Step 1	Map received IP precedence values to internal DSCP values.	**set qos ipprec-dscp-map** *dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8*
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

Enter 8 internal DSCP values to which QoS maps received IP precedence values 0 through 7. This example shows how to map received IP precedence values to internal DSCP values:

```
Console> (enable) set qos ipprec-dscp-map 20 30 1 43 63 12 13 8
QoS ipprec-dscp-map set successfully.
Console> (enable)
```

To revert to default IP precedence to DSCP value mapping, perform this task in privileged mode:

	Task	Command
Step 1	Revert to IP precedence value to DSCP value map defaults.	**clear qos ipprec-dscp-map**
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

This example shows how to revert to QoS map defaults:

```
Console> (enable) clear qos ipprec-dscp-map
QoS ipprec-dscp-map setting restored to default.
Console> (enable)
```

Mapping Internal DSCP Values to Egress CoS Values

To map internal DSCP values to the egress CoS values used for egress port scheduling and congestion avoidance, perform this task in privileged mode:

	Task	Command
Step 1	Map internal DSCP values to egress CoS values.	**set qos dscp-cos-map** *dscp_list:cos ...*
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

For more information, see the "Internal DSCP Values" section on page 41-15 and the "Ethernet Egress Port Scheduling, Congestion Avoidance, and Marking" section on page 41-24.

Enter up to 64 internal DSCP value list/egress CoS value pairs. This example shows how to map internal DSCP values to egress CoS values:

```
Console> (enable) set qos dscp-cos-map 20-25:7 33-38:3
QoS dscp-cos-map set successfully.
Console> (enable)
```

To revert to default CoS to DSCP value mapping, perform this task in privileged mode:

	Task	Command
Step 1	Revert to DSCP value/CoS value map defaults.	**clear qos dscp-cos-map**
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

This example shows how to revert to CoS-DSCP map defaults:

```
Console> (enable) clear qos dscp-cos-map
QoS dscp-cos-map setting restored to default.
Console> (enable)
```

Mapping DSCP Markdown Values

To map DSCP markdown values used by policing rules, perform this task in privileged mode:

	Task	Command
Step 1	Map DSCP values to markdown DSCP values.	**set qos policed-dscp-map** *dscp_list:markdown_dscp ...*
Step 2	With PFC2, map DSCP values to markdown DSCP values.	**set qos policed-dscp-map** [**normal** | **excess**] *in_profile_dscp_list:policed_dscp ...*
Step 3	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

For more information, see the "Policing Rules" section on page 41-22.

Enter up to 64 DSCP-value-list/DSCP-value pairs.

This example shows how to map DSCP markdown values:

```
Console> (enable) set qos policed-dscp-map 20-25:7 33-38:3
QoS dscp-dscp-map set successfully.
Console> (enable)
```

This example shows how to map DSCP markdown values for packets exceeding the excess rate:

```
Console> (enable) set qos policed-dscp-map 33:30
QoS normal-rate policed-dscp-map set successfully.
Console> (enable) set qos policed-dscp-map excess-rate 33:30
QoS excess-rate policed-dscp-map set successfully.
Console> (enable)
```

Note Configure marked-down DSCP values that map to CoS values consistent with the markdown penalty (see the "Mapping Internal DSCP Values to Egress CoS Values" section on page 41-56).

To revert to default DSCP markdown value mapping, perform this task in privileged mode:

	Task	Command
Step 1	Revert to DSCP markdown map defaults.	**clear qos policed-dscp-map** [**normal-rate** | **excess-rate**]
Step 2	Verify the configuration.	**show qos maps** {**config** | **runtime**} [**cos-dscp-map** | **ipprec-dscp-map** | **dscp-cos-map** | **policed-dscp-map**]

This example shows how to revert to DSCP markdown map defaults:

```
Console> (enable) clear qos policed-dscp-map
QoS dscp-cos-map setting restored to default.
Console> (enable)
```

Note Without the **normal-rate** or the **excess-rate** keywords, the **clear qos policed-dscp-map** command clears only the normal policed-dscp map.

Displaying QoS Information

To display QoS information, perform this task:

Task	Command
Display QoS information.	**show qos info** [**runtime** | **config**]

This example shows how to display the QoS runtime information for port 2/1:

```
Console> show qos info config 2/1
QoS setting in NVRAM:
QoS is enabled
Port 2/1 has 2 transmit queue with 2 drop thresholds (2q2t).
Port 2/1 has 1 receive queue with 4 drop thresholds (1q4t).
Interface type:vlan-based
ACL attached:
The qos trust type is set to untrusted.
Default CoS = 0
Queue and Threshold Mapping:
Queue Threshold CoS
----- --------- ---------------
1     1         0 1
1     2         2 3
2     1         4 5
2     2         6 7
Rx drop thresholds:
Rx drop thresholds are disabled for untrusted ports.
Queue #  Thresholds - percentage (abs values )
-------  ---------------------------------------
1        50% 60% 80% 100%
Tx drop thresholds:
Queue #  Thresholds - percentage (abs values )
-------  ---------------------------------------
1        40% 100%
2        40% 100%
```

```
Tx WRED thresholds:
WRED feature is not supported for this port_type.
Queue Sizes:
Queue #  Sizes - percentage (abs values )
-------  ------------------------------------
1        80%
2        20%
WRR Configuration of ports with speed 1000Mbps:
Queue #  Ratios (abs values )
-------  ------------------------------------
1        100
2        255
Console> (enable)
```

Displaying QoS Statistics

To display QoS statistics, perform this task:

Task	Command
Display QoS statistics.	**show qos statistics** {*mod*[/*port*] | **l3stats** | **aggregate-policer** [*policer_name*]}

This example shows how to display QoS statistics for port 2/1:

```
Console> (enable) show qos statistics 2/1
On Transmit:Port 2/1 has 2 Queue(s) 2 Threshold(s)
Q #  Threshold #:Packets dropped
---  ---------------------------------------------
1    1:0 pkts, 2:0 pkts
2    1:0 pkts, 2:0 pkts
On Receive:Port 2/1 has 1 Queue(s) 4 Threshold(s)
Q #  Threshold #:Packets dropped
---  ---------------------------------------------
1    1:0 pkts, 2:0 pkts, 3:0 pkts, 4:0 pkts
```

This example shows how to display QoS Layer 3 statistics:

```
Console> (enable) show qos statistics l3stats
QoS Layer 3 Statistics show statistics since last read.
Packets dropped due to policing: 0
IP packets with ToS changed:     0
IP packets with CoS changed:     26
Non-IP packets with CoS changed: 0
Console>
```

This example shows how to display QoS aggregate policer statistics:

```
Console> (enable) show qos statistics aggregate-policer
QoS aggregate-policer statistics:
Aggregate Policer               Packet Count Packets exceed Packets exceed
                                             normal rate    excess rate
------------------------------- ------------ -------------- -----------------
test                                    1000             20                 5
```

Reverting to QoS Defaults

Note Reverting to defaults disables QoS, because QoS is disabled by default.

To revert to QoS defaults, perform this task in privileged mode:

Task	Command
Revert to QoS defaults.	**clear qos config**

This example shows how to revert to QoS defaults:

```
Console> (enable) clear qos config
This command will disable QoS and take values back to factory default.
Do you want to continue (y/n) [n]? y
QoS config cleared.
Console> (enable)
```

Disabling QoS

To disable QoS, perform this task in privileged mode:

Task	Command
Disable QoS on the switch.	**set qos** {**enable** | **disable**}

This example shows how to disable QoS:

```
Console> (enable) set qos disable
QoS is disabled.
Console> (enable)
```

Configuring COPS Support

Note The commands in this section are not supported with a Layer 2 Switching Engine.

Note COPS can configure QoS only for IP traffic. Use the CLI or SNMP to configure QoS for all other traffic.

These sections describe configuring COPS support:

Note Throughout this publication and all Catalyst 6000 family documents, the term “COPS” refers to COPS support as implemented on the Catalyst 6000 family switches.

Port ASICs

Some COPS support features affect all ports controlled by a port ASIC. The following sections use the term “per-ASIC” to identify features that configure all ports on the same port ASIC:

- The port ASICs on Gigabit Ethernet switching modules control up to 4 ports each: 1–4, 5–8, 9–12, and 13–16.
- There is a port ASIC on 10-Mbps, 10/100-Mbps, and 100-Mbps Ethernet switching modules that controls all ports.
- On 10-Mbps, 10/100-Mbps, and 100-Mbps Ethernet switching modules, there is another set of port ASICs that control 12 ports each (1–12, 13–24, 25–36, and 37–48), but COPS cannot configure them.
- Changes to an EtherChannel port apply to all ports in the EtherChannel and to all ports controlled by the ASIC (or ASICs) that control the EtherChannel ports.

Understanding QoS Policy

The term *QoS policy* refers to the QoS values in effect, such as port trust state and which ACLs are applied to ports and VLANs.

Selecting COPS as the QoS Policy Source

QoS uses locally configured QoS values as the default QoS policy source. To select COPS as the QoS policy source, perform this task in privileged mode:

	Task	Command
Step 1	Select COPS as the QoS policy source.	**set qos policy-source** {**local** | **cops**}
Step 2	Verify the QoS policy source.	**show qos policy-source**

This example shows how to select COPS as the QoS policy source:

```
Console> (enable) set qos policy-source cops
QoS policy source for the switch set to COPS.
Console> (enable) show qos policy-source
QoS policy source for the switch set to COPS.
Console> (enable)
```

Selecting COPS as the QoS policy source switches the following values from locally configured values to received COPS values:

- All DSCP maps
- Named and default ACL definitions
- Microflow and aggregate policing rules
- CoS to queue assignments
- Threshold configuration
- WRR weight and buffer configuration
- Default port CoS and ACL-to-interface attachments

Selecting Locally Configured QoS Policy

To select locally configured QoS policy, perform this task in privileged mode:

	Task	Command
Step 1	Select locally configured QoS policy.	**set qos policy-source** {**local** | **cops**}
Step 2	Verify the QoS policy source.	**show qos policy-source**

This example shows how to select locally configured QoS policy:

```
Console> (enable) set qos policy-source local
QoS policy source for the switch set to local.
Console> (enable) show qos policy-source
QoS policy source for the switch set to local.
Console> (enable)
```

Enabling Use of Locally Configured QoS Policy

When enabled, COPS is the default QoS policy source for all ports. You can use locally configured QoS policy on a per-ASIC basis. To enable use of locally configured QoS policy on a port ASIC, perform this task in privileged mode:

	Task	Command
Step 1	Enable use of locally configured QoS policy on a port.	**set port qos policy-source** {**local** | **cops**}
Step 2	Verify the QoS policy source for the port.	**show port qos**

This example shows how to enable use of locally configured QoS policy:

```
Console> (enable) set port qos 1/1 policy-source local
QoS policy source set to local on port(s) 1/1-2.
Console> (enable)
```

Assigning Port Roles

COPS does not configure ports using slot number and port number parameters. COPS uses *roles* that you create and assign to port ASICs.

A role is a name that describes the capability of ports (for example, *access* or *mod2_1-4*). QoS supports 64 roles per switch. You can assign more than one role to a port ASIC (for example, *mod2ports1-12* and *access*), with the limitation that the combined length of role names assigned to a port ASIC cannot exceed 255 characters.

The role name can be up to 31 characters long, is not case sensitive but may include uppercase and lowercase characters, and may consist of a–z, A–Z, 0–9, the dash character (-), the underscore character (_), and the period character (.). Role names cannot start with the underscore character.

The first assignment of a new role to a port creates the role.

To assign roles to a port ASIC, perform this task in privileged mode:

	Task	Command
Step 1	Assign roles to a port ASIC.	**set port cops** {*mod/port*} **roles** *role1* [*role2*] ...
Step 2	Verify the roles for the port.	**show port cops** [*mod*[*/port*]]

This example shows how to assign two new roles to the ASIC controlling port 2/1:

```
Console> (enable) set port cops 2/1 roles mod2ports1-12 access
New role 'mod2ports1-12' created.
New role 'access' created.
Roles added for port 2/1-12.
Console> (enable)
```

Removing Roles from Port ASICs

To remove a role from a port ASIC, perform this task in privileged mode:

	Task	Command
Step 1	Remove a role from a port ASIC.	**clear port cops** {*mod/port*} {**all-roles** | **roles** *role1* [*role2*] ...}
Step 2	Verify the roles for the port.	**show port cops** [*mod*[*/port*]]

This example shows how to remove a role from a port ASIC:

```
Console> (enable) clear port cops 3/1 roles backbone_port main_port
Roles cleared for port(s) 3/1-4.
Console> (enable)
```

Deleting Roles

To delete a role (which removes it from all ports), perform this task in privileged mode:

	Task	Command
Step 1	Delete a role.	**clear cops** {**all-roles** | **roles** *role1* [*role2*] ...}
Step 2	Verify the roles for the port.	**show port cops** [*mod*[*/port*]]

This example shows how to delete a role:

```
Console> (enable) clear cops roles backbone_port main_port
Roles cleared.
Console> (enable)
```

Configuring Policy Decision Point Servers

Note COPS and RSVP can use the same policy decision point (PDP) server.

COPS obtains QoS policy from a PDP server. Configure a primary PDP server and, optionally, a backup PDP server.

To configure a PDP server, perform this task in privileged mode:

	Task	Command
Step 1	Configure a PDP server.	**set cops server** *ip_address* [*port*] [**primary**] [**diff-serv** | **rsvp**]
Step 2	Verify the PDP server configuration.	**show cops info**

The *ip_address* parameter can be the IP address or name of the server.

The *port* variable is the PDP server TCP port number.

Use the **diff-serv** keyword to set the address only for COPS.

This example shows how to configure a PDP server:

```
Console> (enable) set cops server my_server1 primary
my_server1 added to the COPS diff-serv server table as primary server.
my_server1 added to the COPS rsvp server table as primary server.
Console> (enable)
```

Deleting PDP Server Configuration

To delete PDP server configuration, perform this task in privileged mode:

	Task	Command
Step 1	Delete PDP server configuration.	**clear cops server** {**all** | *ip_address* [**diff-serv** | **rsvp**]}
Step 2	Verify the PDP server configuration.	**show cops info**

This example shows how to delete PDP server configuration:

```
Console> (enable) clear cops server all
All COPS diff-serv servers cleared.
All COPS rsvp servers cleared.
Console> (enable)
```

Configuring the COPS Domain Name

PDP servers use a COPS domain name to communicate with policy enforcement point (PEP) devices such as switches. To configure a COPS domain name for the switch, perform this task in privileged mode:

	Task	Command
Step 1	Configure the COPS domain name.	**set cops domain-name** *domain_name*
Step 2	Verify the COPS domain name.	**show cops info**

This example shows how to configure a COPS domain name:

```
Console> (enable) set cops domain-name my_domain
Domain name set to my_domain.
Console> (enable)
```

Deleting the COPS Domain Name

To delete the COPS domain name, perform this task in privileged mode:

	Task	Command
Step 1	Delete the COPS domain name.	**clear cops domain-name**
Step 2	Verify the configuration.	**show cops info**

This example shows how to delete the COPS domain name:

```
Console> (enable) clear cops domain-name
Domain name cleared.
Console> (enable)
```

Configuring the COPS Communications Parameters

To configure the parameters COPS uses to communicate with the PDP server, perform this task in privileged mode:

	Task	Command
Step 1	Configure the parameters COPS uses to communicate with the PDP server.	**set cops retry-interval** *initial increment maximum*
Step 2	Verify the configuration.	**show cops info**

Enter the parameters as a number of seconds in the range 0 to 65535. The value of the *initial* parameter plus the value of the *increment* parameter must not exceed the value of the *maximum* parameter.

This example shows how to configure the parameters COPS uses to communicate with the PDP server:

```
Console> (enable) set cops retry-interval 15 1 30
Connection retry intervals set.
Console> (enable)
```

Configuring RSVP Support

Note The commands in this section are not supported with a Layer 2 Switching Engine.

These sections describe configuring RSVP null service template and receiver proxy functionality support:

Note Throughout this publication and all Catalyst 6000 family switch documents, the term “RSVP” refers to RSVP null service template and receiver proxy functionality support as implemented on the Catalyst 6000 family switches.

Enabling RSVP Support

To enable RSVP support, perform this task in privileged mode:

	Task	Command
Step 1	Enable RSVP support on the switch.	**set qos rsvp** {**enable** | **disable**}
Step 2	Verify the configuration.	**show qos rsvp info**
Step 3	Display RSVP activity.	**show qos rsvp flow-info**

This example shows how to enable RSVP support:

```
Console> (enable) set qos rsvp enable
RSVP enabled on the switch.
Console> (enable)
```

Disabling RSVP Support

To disable RSVP support, perform this task in privileged mode:

	Task	Command
Step 1	Disable RSVP support on the switch.	**set qos rsvp** {**enable** | **disable**}
Step 2	Verify the configuration.	**show qos rsvp info**

This example shows how to disable RSVP support:

```
Console> (enable) set qos rsvp disable
RSVP disabled on the switch.
Console> (enable)
```

Enabling Participation in the DSBM Election

Catalyst 6000 family switches can serve as the Designated Subnet Bandwidth Manager (DSBM). You can enable participation in the election of the DSBM on a per-port basis.

Note The DSBM is not reelected when additional RSVP devices join the network. To control which device is the DSBM, disable election participation in all devices except the one that you want elected as DSBM. After the DSBM is elected, reenable election participation in other devices, as appropriate for the network configuration.

To enable the participation of a port in the election of the DSBM, perform this task in privileged mode:

	Task	Command
Step 1	Enable the participation of a port in the election of the DSBM.	**set port rsvp** {*mod/port*} **dsbm-election** {**disable** | **enable** *priority*}
Step 2	Verify the configuration of the port.	**show port rsvp** [*mod*[*/port*]

The range for the *priority* parameter is 128 to 255.

This example shows how to enable the participation of ports 2/1 and 3/2 in the election of the DSBM:

```
Console> (enable) set port rsvp 2/1,3/2 dsbm-election enable 232
DSBM enabled and priority set to 232 for ports 2/1,3/2.
Console> (enable)
```

Disabling Participation in the DSBM Election

To disable the participation of a port in the election of the DSBM, perform this task in privileged mode:

	Task	Command
Step 1	Disable the participation of a port in the election of the DSBM.	**set port rsvp** {*mod/port*} **dsbm-election** {**disable** | **enable** *priority*}
Step 2	Verify the configuration.	**show port rsvp show port rsvp** [*mod*[*/port*]]

This example shows how to disable the participation of port 2/1 in the election of the DSBM:

```
Console> (enable) set port rsvp 2/1 dsbm-election disable
DSBM disabled for port 2/1.
Console> (enable)
```

Configuring Policy Decision Point Servers

Note COPS and RSVP can use the same PDP server.

When the switch is the DSBM, RSVP communicates with a PDP server. Configure a primary PDP server and, optionally, a backup PDP server.

To configure a PDP server, perform this task in privileged mode:

	Task	Command
Step 1	Configure a PDP server.	**set cops server** *ip_address* [*port*] [**primary**] [**diff-serv** | **rsvp**]
Step 2	Verify the PDP server configuration.	**show cops info**

The *ip_address* parameter can be the IP address or name of the server.

The *port* variable is the PDP server TCP port number.

Use the **rsvp** keyword to set the address only for RSVP.

This example shows how to configure a PDP server:

```
Console> (enable) set cops server my_server1 primary rsvp
my_server1 added to the COPS rsvp server table as primary server.
Console> (enable)
```

Deleting PDP Server Configuration

To delete PDP server configuration, perform this task in privileged mode:

	Task	Command
Step 1	Delete PDP server configuration.	**clear cops server** {**all** | *ip_address* [**diff-serv** | **rsvp**]}
Step 2	Verify the PDP server configuration.	**show cops info**

Use the **rsvp** keyword to delete only the RSVP address.

This example shows how to delete PDP server configuration:

```
Console> (enable) clear cops server all
All COPS diff-serv servers cleared.
All COPS rsvp servers cleared.
Console> (enable)
```

Configuring RSVP Policy Timeout

When the switch is the DSBM and communication with the PDP server is lost, the switch continues to function as the DSBM, using cached values, for the period specified by the timeout value; the behavior for new or modified RSVP **path** messages is determined by the RSVP local policy setting.

If communication with the PDP server is not reestablished before the timeout period expires, the switch reverts to the role of Subnet Bandwidth Manager (SBM) client for all ports and forwards RSVP messages to a newly elected DSBM on the segment. When there is no communication with the PDP server, the switch does not participate in election of the DSBM.

To configure the time that the switch continues to be the DSBM after communication with the PDP server is lost, perform this task in privileged mode:

	Task	Command
Step 1	Configure the RSVP policy timeout.	**set qos rsvp policy-timeout** *timeout*
Step 2	Verify the configuration.	**show qos rsvp info**

Enter the *timeout* parameter as a number of minutes in the range 0 to 65535 (default is 30).

This example shows how to configure the RSVP policy timeout:

```
Console> (enable) set qos rsvp policy-timeout 45
RSVP database policy timeout set to 45 minutes.
Console> (enable)
```

Configuring RSVP Use of Local Policy

To configure how RSVP operates after communication with the PDP is lost, perform this task in privileged mode:

	Task	Command
Step 1	Configure how RSVP operates when there is no communication with the PDP server.	**set qos rsvp local-policy** {**forward** | **reject**}
Step 2	Verify the configuration.	**show qos rsvp info**

The **forward** keyword sets the local policy to forward all new or modified RSVP **path** messages. The **reject** keyword sets the local policy to reject all new or modified RSVP **path** messages. This example shows how to change the default local RSVP policy setting to reject all new or modified RSVP **path** messages:

```
Console> (enable) set qos rsvp local-policy reject
RSVP local policy set to reject.
Console> (enable)
```

Note The RSVP local policy is only used until the RSVP policy timeout expires after the connection to the PDP is lost. After the RSVP policy timeout expires, the switch behaves as an SBM client. RSVP messages pass through the switch unchanged regardless of the RSVP local policy setting. The RSVP local policy setting is not used if the switch never establishes a connection to the PDP.

Configuring QoS Statistics Data Export

These sections describe how to configure the QoS statistics data export feature:

Enabling QoS Statistics Data Export Globally

To export QoS statistics data for ports and aggregate policers, you must first configure the feature globally.

To enable QoS statistics data export globally, perform this task in privileged mode:

	Task	Command
Step 1	Enable QoS statistics data export.	**set qos statistics export enable | disable**
Step 2	Verify the configuration.	**show qos statistics export info**

This example shows how to enable QoS statistics data export globally and verify the configuration:

```
Console> (enable) set qos statistics export enable
Export is enabled.
Export destination:172.20.52.3 SYSLOG facility LOG_LOCAL6 (176), severity LOG_DE
BUG (7)
Aggregate policer export is not supported
Console> (enable) show qos statistics export info
Statistics export status and configuration information
-----------------------------------------------------
Export status: enabled
Export time interval: 300
Export destination:172.20.52.3 SYSLOG facility LOG_LOCAL6 (176), severity LOG_DE
BUG (7)
Port    Export
------  --------
 1/1    disabled
 1/2    disabled
 3/1    disabled
 3/2    disabled
 5/1    disabled
 5/2    disabled
 5/3    disabled
 5/4    disabled
<output truncated>
Console> (enable)
```

Enabling Per-Port QoS Statistics Data Export

To enable QoS statistics data export on a per-port basis, perform this task in privileged mode:

	Task	Command
Step 1	Enable QoS statistics data export per port.	**set qos statistics export port** *mod/port* **enable | disable**
Step 2	Verify the configuration.	**show qos statistics export info**

Note You must enable QoS statistics data export globally in order for the per-port configuration to take effect.

This example shows how to enable the QoS statistics data export feature per port and verify the configuration:

```
Console> (enable) set qos statistics export port 5/1 enable
Port export enabled on 5/1.
Console> (enable) show qos statistics export info
Statistics export status and configuration information
-----------------------------------------------------
Export status: enabled
Export time interval: 300
Export destination:172.20.52.3 SYSLOG facility LOG_LOCAL6 (176), severity LOG_DE
BUG (7)
Port    Export
------  --------
 1/1    disabled
 1/2    disabled
 3/1    disabled
 3/2    disabled
 5/1    enabled
 5/2    disabled
<output truncated>
Console> (enable)
```

When enabled on a port, QoS statistics data export contains the following fields, separated by the delimiter character:

- Export type ("1" for a port)
- Slot/port
- Number of ingress packets
- Number of ingress bytes
- Number of egress packets
- Number of egress bytes
- Time stamp

Enabling Per-Aggregate Policer QoS Statistics Data Export

To enable QoS statistics data export on a per-aggregate policer basis, perform this task in privileged mode:

	Task	Command
Step 1	Enable QoS statistics data export per aggregate policer.	**set qos statistics export enable | disable**
Step 2	Verify the configuration.	**show qos statistics export info**

Note You must enable QoS statistics data export globally in order for the per-aggregate policer configuration to take effect.

This example shows how to enable QoS statistics data export for a specific aggregate policer and verify the configuration:

```
Console> (enable) set qos statistics export aggregate ipagg_3 enable
Statistics data export enabled for aggregate policer ipagg_3
Console> (enable) show qos statistics export info
Statistics export status and configuration information
-----------------------------------------------------
Export status: enabled
Export time interval: 300
Export destination:172.20.52.3 SYSLOG facility LOG_LOCAL6 (176), severity LOG_DE
BUG (7)
Port    Export
------  --------
 1/1    disabled
 1/2    disabled
 3/1    disabled
 3/2    disabled
 5/1    enabled
 5/2    disabled
<output truncated>

Aggregate name  Export
--------------  --------
ipagg_3         enabled
Console> (enable)
```

When enabled for a named aggregate policer, QoS statistics data export contains the following fields, separated by the delimiter character:

- Export type ("3" for an aggregate policer)
- Aggregate policer name
- Number of in-profile packets
- Number of packets that exceed the CIR
- Number of packets that exceed the PIR
- Time stamp

Setting the QoS Statistics Data Export Time Interval

The default interval at which QoS statistics is exported is 30 seconds. To set the time interval for the QoS statistics data export, perform this task in privileged mode:

	Task	Command
Step 1	Set the time interval for the QoS statistics data export.	**set qos statistics export interval** *interval*
Step 2	Verify the configuration.	**show qos statistics export info**

This example shows how to set the QoS statistics data export interval and verify the configuration:

```
Console> (enable) set qos statistics export interval 500
Time interval set to 500
Console> (enable) show qos statistics export info
Statistics export status and configuration information
-----------------------------------------------------
Export status: enabled
Export time interval: 500
Export destination:172.20.52.3 SYSLOG facility LOG_LOCAL6 (176), severity LOG_DE
BUG (7)
Port    Export
------  --------
 1/1    disabled
 1/2    disabled
 3/1    disabled
 3/2    disabled
 5/1    enabled
 5/2    disabled
<output truncated>

Aggregate name  Export
--------------  --------
ipagg_3         enabled
Console> (enable)
```

Configuring QoS Statistics Data Export Destination Host and UDP Port

To configure the QoS statistics data export destination host and UDP port number, perform this task in privileged mode:

	Task	Command
Step 1	Configure the QoS statistics data export destination host and UDP port number.	**set qos statistics export destination** {*host_name* | *ip_address*} [**syslog** [*facility* | *severity*] *port*]
Step 2	Verify the configuration.	**show qos statistics export info**

This example shows how to configure the QoS statistics data export destination host and UDP port number and verify the configuration:

```
Console> (enable) set qos statistics export destination stargate 9996
Statistics data export destination set to stargate port 9996.
Console> (enable) show qos statistics export info
Statistics export status and configuration information
-----------------------------------------------------
Export status: enabled
```

```
Export time interval: 500
Export destination:Stargate, UDP port 9996
Port    Export
------  --------
 1/1    disabled
 1/2    disabled
 3/1    disabled
 3/2    disabled
 5/1    enabled
 5/2    disabled
<output truncated>

Aggregate name  Export
--------------  --------
ipagg_3         enabled
Console> (enable)
```

Displaying QoS Statistics Information

To display the per-port and per-aggregate policer packet and byte rates, perform this task in privileged mode:

Task	Command
Display the per port and per aggregate policer packet and byte rates.	**show mac** [**utilization**] [*mod*[/*port*]

This example shows how to display theQoS statistics per-port and per-aggregate policer packet and byte rates:

```
Console> (enable) show mac utilization 1

5 min input/output port rates:

Port  Xmit-Packet-Rate     Xmit-Octet-Rate
----- -------------------- --------------------
1/1                  1343               123432
1/2                  2342               232343
Port  Rcv-Packet-Rate      Rcv-Octet-Rate
----- -------------------- --------------------
1/1                  1324               143253
1/2                  2234               253234
Console>(enable)
```